AN EXPLORATORY JOURNEY OF REPRESENTATION LEARNING'S ENHANCEMENT, ADAPTATION AND RELATED INTELLIGENT METHODS

BY

JING WU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2024

Urbana, Illinois

Doctoral Committee:

Professor Naira Hovakimyan, Chair
Professor Srinivasa Salapaka
Associate Professor Nicolas Martin
Assistant Professor Yuxiong Wang

Abstract

Representation learning models employing Siamese structures have consistently demonstrated exceptional performance across various fields, including deep learning, computer vision, and natural language processing. Furthermore, the applicability of representation learning has broadened to encompass wider domains such as agriculture, remote sensing, and earth observation, which are significantly challenged by data scarcity. This dissertation aims to enhance the quality and adaptability of learned representations across these diverse application domains. Meanwhile, we have also expanded the scope of our research to a broader area of intelligent agricultural systems.

Initially, we delve into contrastive representation learning within the general computer vision domain and introduce a novel "Hallucinator" module to reduce mutual information, increase the batch size of positive pairs, and improve representation quality. Subsequently, we extend the representation framework to agriculture and remote sensing, proposing spatial-temporal-aware architectures tailored to the unique characteristics of remote sensing data. Furthermore, we introduce the Extended Agriculture Vision dataset to address data scarcity issues and showcase the effectiveness of proposed representation frameworks.

Furthermore, we demonstrate that the learned representations are powerful features for few-shot tasks in remote sensing and earth observation. We introduce GenCo, a generator-based representation learning

framework that simultaneously pre-trains backbones and explores variants of feature samples. During fine-tuning, the auxiliary generator enriches the limited labeled data samples in the feature space. We validate the effectiveness of our method in enhancing few-shot learning performance on the Agriculture-Vision and EuroSAT datasets. Notably, our few-shot approach surpasses purely supervised training in both classification and semantic segmentation tasks trained over ten thousand images in the Agriculture-Vision Dataset.

Lastly, we propose an intelligent nitrogen (N) management system utilizing deep reinforcement learning (RL) in conjunction with crop simulations through the Decision Support System for Agrotechnology Transfer (DSSAT). Initially, we framed the N management issue as an RL problem. Subsequently, we train management policies using deep Q-network and soft actor-critic algorithms, along with the Gym-DSSAT interface. This interface facilitates daily interactions between the simulated crop environment and RL agents. According to our experiments with maize crops in both Iowa and Florida, USA, the RL-trained policies surpass previous empirical methods.

To My Dear Parents

人归落雁后
思发在花前

Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor Naira Hovakimyan, for her unwavering patience, guidance, and steadfast support throughout my academic journey at UIUC. The work summarized in this thesis would not be possible without her.

I extend my sincere thanks to my committee members, Professor Nicolas Martin, Professor Srinivasa Salapaka, and Professor Yuxiong Wang, for generously dedicating their time, commitment, and providing invaluable constructive feedback.

It has been a great honor to work with many excellent researchers during my internships at Amazon, including Dr. Suiyao Chen, Dr. Qi Zhao, Dr. Chongchao Zhao, and Dr. Daniel Cociorva. Especially, I am indebted to my mentor, Dr. Jennifer Hobbs, at Intelinair, for her insightful suggestions, guidance, and unwavering encouragement during my Intelinair internships.

To all my collaborators, Yite and Shengjie, your dedication and the invaluable research insights you've shared have enriched my work with diverse perspectives, and for that, I am immensely grateful.

To the past and present members of the Advanced Control Research Lab – Pan, Sheng, Tigran, Aditya, Andrew, Arun, Chuyuan, Christoph, Chengyu, Hunmin, Donglei, Hyungjin, Hyungsoo, John, Lin, Mikayel, Minjun, Minkyung, Neng, Ran, Vivek, Yanbing, Yikun, Yuliang, Zhuohuan, Ziyao – I extend my heartfelt

appreciation for our stimulating research discussions and the cherished memories we have created together.

Lastly, to my family and friends, including Yulei, Tangyuan, Yining, Min, Zhongyi, Yite and Jianjia, your unwavering support and boundless love have been my anchor throughout this journey. Thank you for being my rock.

Table of contents

List of Abbreviations and Symbols

$\mathbb{R}^n$	N-dimensional Euclidean space.
$\|.\|$	Absolute value.
$\|\|.\|\|$	The norm of a vector.
U	Uniform distribution.
β	Beta distribution.
AV	Agriculture-Vision dataset.
AV+	Extended Agriculture-Vision dataset.
SSL	Unsupervised and self-supervised learning.
NIR	Near-infrared.
PPM	Pixel-to-Propagation Module.
TemCo	Temporal Contrast.
Genco	Generator-based contrastive learning framework.
EO	Earth observation.
RS	Remote sensing.
GAN	Generative Adversarial Network.
ADAPT	bAlanced DynAmic sParse Training.
DST	Dynamic sparse training.
NAS	Neural architecture search.
N	Nitrogen.
RL	Reinforcement Learning.
DQN	Q-network
SAC	Soft actor-critic algorithms
LLM	Large language model.

DSSAT	Decision Support System for Agrotechnology Transfer.
DNNs	Deep neural networks.
MDP	Markov Decision Process.
$H(.)$	Hallucinator.
Cls	Classification task.
Seg	Semantic Segmentation task.
$\mathcal{L}$	Loss function.
$S(\cdot, \cdot)$	Similarity function.

Chapter 1

Introduction

In the recent computer vision community, there has been rapid progress in self-supervised learning (SSL), gradually closing the performance gap with supervised learning [1]–[5]. Among the diverse approaches of SSL, contrastive learning methods, such as MoCoV1&V2 [2], [6], SimCLR [3], and SimSiam [7], show promising results. Generally, contrastive learning treats each image as one class which will be augmented into two separate views. These two views form one positive pair and should ideally be close if mapped to feature space. With sufficient contrast in the feature space, contrastive learning models show a strong capacity to learn transformation-invariant features that are transferable to various downstream tasks, such as classification, object detection, and segmentation [8]–[10].

To ensure sufficient contrast, researchers from previous work address the issue from two essential practices, either introducing large amounts of positive pairs or adding additional variants&transformation among them. For example, SimCLR uses a batch size that generates thousands of positives to facilitate the convergence of models [3]. Other research reduces mutual information of positive pairs using stronger data augmentation, i.e., color distortion and jigsaw transformation [5], [11]. Likewise, the work from [12]

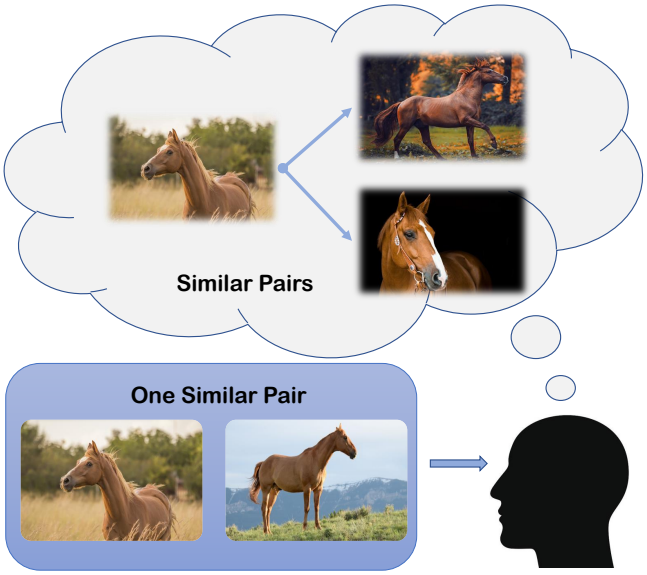

Figure 1.1: The motivation of the proposed hallucination methods. Given one pair of images with the same semantic meaning, such as a pair of horses, a person can envision further similar pairs by imagining one of the horses in different poses and surroundings. If a contrastive learning model could do such hallucination, it could have additional novel pairs to contrast given the same data. Note that this hallucination process is for illustration only. In the implementation, all the hallucinated samples are computed in the feature space.

introduces ContrastiveCrop, and the work from [13] proposes Un-Mix, respectively, to reduce the similar semantic meaning of sample pairs in the original image space. Beyond the data augmentation and image operations, researchers from [14] propose to apply a linear operation to generate hard positive samples in feature space.

Despite the success of prior approaches, we argue that large batch sizes are not always achievable. Meanwhile, all proposed techniques only focus on improving the original pairs. Given one positive pair of positive samples, humans are born with the amazing ability to come up with additional positives by imagining a sample from different surroundings and perspectives without much effort, as demonstrated in Figure 1.1. This process of self-imagination, in turn, will benefit the human neurological system, improving recognition capacity [15]. Similarly, if we could empower contrastive learning models with the ability

to hallucinate or imagine an object to a novel view, additional positive pairs could be provided for the learning tasks.

Unfortunately, exploring feasible methods to hallucinate novel positive pairs is challenging. Firstly, while generative models produce realistic images that could form additional positive views [16]–[19], realistic data do not necessarily benefit learning tasks [20]. More importantly, applying these approaches forces us to fall back into a computational dilemma to the previous method. In other words, image-level hallucination still suffers from expensive computation as we still need to encode the hallucinated images into feature space. Lastly, if the generated positive pairs are similar to each other, training a discriminative model would be too trivial, thus showing poor generalization capacity [5], [12], [14].

Therefore, the key insight of the dissertation is that the sample-generation process should aim for three critical elements: (i) feature-space operation, (ii) sufficient variance of positive pairs, (iii) a differentiable module optimized directly related to the learning task. To achieve this, we propose *Hallucinator* to improve the performance of contrastive learning with Siamese structures. The *Hallucinator* is plugged in after the encoder to manipulate feature vectors and improve the feature-level batch size for further contrast. To ensure adequate variance is introduced, we propose an asymmetric feature extrapolation method inspired by the work from [14]. More importantly, we present a non-linear hallucination process for the extrapolated samples. Such a process is differentiable (i.e., learnable), therefore essentially boosting *Hallucinator* to generate smooth and task-related features.

The proposed *Hallucinator* delivers extra positives and simultaneously enlarges the variance between newly introduced pairs. Moreover, this approach only relies on positive samples. Therefore, it can be easily applied to any Siamese structure by adding it after the encoders as a plug-and-play module. Without the tedious exploration of hyper-parameters and much additional computation, we empirically prove the

effectiveness of the proposed *Hallucinator* on popular contrastive learning models, including MoCoV1&V2, SimCLR and SimSiam. We notice a stable improvement ranging from 0.3% to 3.0% under the linear classification protocol, crossing the CIFAR10&100, Tiny ImageNet, STL-10 and ImageNet. We also observe that models trained with *Hallucinator* show better transferability in downstream tasks like object detection and segmentation.

In light of the success of representation learning in computer vision for addressing the need for large labeled datasets, another key insight of this dissertation lies in expanding its power in broader application domains, including agriculture and remote sensing.

While massive annotated datasets like ImageNet have fostered the development of powerful and robust deep-learning models for natural images [21]–[25], creating large, complex datasets is costly, time-consuming, and may be infeasible in domains like remote sensing and agriculture. To be more specific, obtaining large quantities of accurate annotations is especially challenging for remote sensing tasks, particularly for agriculture, as objects of interest tend to be very small, high in number (perhaps thousands per image), possess complex organic boundaries, and may require channels beyond red-green-blue (RGB) to identify.

Approaches developed initially for natural images may work well on remote sensing imagery with only minimal modification. However, this is not guaranteed due to the large domain gap. Additionally, initial methods may fail to exploit the unique structure of earth observation data, such as geographic consistency or seasonality [26]. Explicitly benchmarking approaches on domain-relevant data is critical.

In this dissertation, we focus on the Agriculture-Vision (AV) dataset [27]: a large, multi-spectral, high-resolution (10 cm/pixel), labeled remote sensing dataset for semantic segmentation. Unlike low-resolution public satellite data, this imagery enables within-field identification of key agronomic patterns such as weeds and nutrient deficiency.

While this dataset is noted for its size, most aerial agriculture datasets are quite small. Therefore, we leverage the large amounts of *un-annotated data* which is readily available in this domain, benchmark several representation learning approaches whose inductive bias reflects the structure of this data, and evaluate the impact of these approaches in more data-limited settings.

Meanwhile, inspired by human's highly efficient learning ability, research around learning from unlabeled data, i.e. unsupervised or self-supervised learning [28], and the ability to generalize from only a few examples, i.e. few-shot learning, have become key areas of interest in the machine learning community [29]–[35]. Few-shot learning aims to realize the knowledge adaptation of embeddings from label-abundant data to label-scarce classes. While the adapted representation aims to discriminate different levels of information (e.g., instance level and semantic level) between classes, the embedding should be invariant to common, irrelevant variations of the image, including different sizes, deformations, and lighting. The question is then: How can we learn a representation invariant to common factors while maintaining differences for diverse classes with limited labels in the Remote sensing (RS) and earth observation (EO) domain?

As raw RS-EO data is highly abundant, but ground-truth data is extremely scarce, leveraging contrastive methods for few-shot learning offers a key opportunity in this domain. Additionally, most common contrastive learning and few-shot methods were developed for natural scene imagery; e.g., [1]–[3], [36] show that as the statistics of that domain (both source imagery and targets) are extremely different from RS-EO data, there is no guarantee that the same benefit will be observed without adaptation. Therefore, we investigate the improvement in the performance of few-shot learning in RS-EO classification and semantic segmentation tasks using contrastive-learning-based pre-training.

Specifically, we focus on pre-training from the Extended Agriculture Vision dataset (AV+) [37], which includes high-resolution aerial imagery over agricultural lands in the US Midwest. Obtaining ground-truth

annotations for agriculture is particularly challenging due to patterns of interest being small in size, high in number, and often possessing ambiguous boundaries; the ability to identify patterns from only a small number of samples addresses key challenges in precision agriculture and food security.

Drawing inspiration from the work of [32], [38], [39], we have adopted a two-stage training approach that involves contrastive-learning-based pre-training followed by fine-tuning. Specifically, we have developed a contrastive learning framework, GenCo, with an auxiliary generator trained. During the pre-training, the generator is tasked with exploring variants of encoded features and formulating additional positive pairs. During the fine-tuning stage, we fixed the parameters of the encoders and trained only the classification layer and decoder for classification and segmentation, respectively. Notably, only a limited number of labeled data samples are provided during the fine-tuning stage. To address this, we introduced the generator from the contrastive learning model to create further samples in feature space and enrich knowledge during the few-shot tasks.

We find that the embeddings pre-trained from AV+ under this protocol show better adaptability when compared to counterparts pre-trained on ImageNet [40] and COCO [10]. Our method outperforms pre-trained ImageNet weights by 1 to 6 points on Agriculture-Vision and EuroSAT classification tasks under the same number of supervised training samples. Similarly, our embeddings deliver a 5 to 7-point improvement on mIoU compared with embeddings learned from COCO on the Agriculture-Vision semantic segmentation task.

Meanwhile, we demonstrate the high learning efficiency of the proposed method for RS-EO imagery. With a few labeled images, we find that the GenCo shows comparable or even better results under different tasks and datasets when compared with supervised models trained on fully labeled data samples; our proposed approach shows matching performance with less than 0.01 percent of labeled data.

This dissertation also extends into practical applications in a vital field, i.e., agriculture management. It highlights the pressing challenges in the agricultural industry, notably the need to meet the increasing food demand for a growing global population projected to exceed nine billion by 2050. These challenges are exacerbated by limited land and water resources, soil degradation, and climate change. Among various factors, N management emerges as a crucial aspect, significantly impacting crop production and the environment. Nitrogen, vital for crop growth and yield, can have adverse environmental effects when used excessively. Consequently, effective N management is critical for optimizing crop yields, increasing farmer income, and minimizing environmental harm. However, there's uncertainty about the optimality of existing N management practices among farmers, especially under adverse seasonal conditions. N management is essentially a sequential decision-making problem, requiring precise decisions on nitrogen application throughout the crop growth cycle.

In response to these agricultural challenges, the dissertation proposes an innovative framework for optimizing N management using deep RL and crop simulations. This approach leverages the DSSAT for crop modeling and the Gym-DSSAT interface, allowing for detailed simulation and interaction with RL agents. By training N management policies using DQN and soft SAC for maize crops in Iowa and Florida, the framework demonstrates significant advancements over standard practices. This approach represents a major step forward compared to earlier RL-based crop management research, offering a more comprehensive and scalable solution with larger state and action spaces. Unlike previous studies, the widely-used DSSAT model underpins this research, enhancing its global applicability. The study also includes an extensive experimental analysis covering multiple deep RL algorithms, geographic locations, and scenarios such as partial observations and reduced action frequencies, showcasing its robustness and comprehensive nature in addressing modern agricultural challenges.

Continuing, we introduce an intelligent crop management framework utilizing deep RL and DSSAT for crop simulations using more recent techniques. This approach is novel in its use of LMs to transform simulation data into descriptive sentences, enhancing the RL agent's understanding of complex crop dynamics. The effectiveness of this LM-based approach is demonstrated in case studies on maize crops in Florida and Zaragoza. The LM-based RL agents outperform traditional methods, offering improvements in crop yield, resource utilization, and environmental impact. This pioneering research paves the way for more advanced, sustainable agricultural practices and contributes significantly to addressing global food security challenges.

1.1 Dissertation Outline

This dissertation has six chapters for which a brief overview is given below:

- Chapter 2 explores the nature of representation learning. The research seeks to pinpoint the crucial factors underpinning the success of various contrastive representation learning models. Our objective is to utilize these identified factors to bolster the robustness and generalization capacity of representations. Ideally, this enhancement should be achieved without adding extra computational overhead and be applicable across a spectrum of contrastive learning models with the Siamese structure.

- Chapter 3 focuses on the creation of remote sensing datasets for agriculture patterns as well as the application of self-supervised benchmarking. These efforts on both fronts enable us to better detect key agricultural patterns of interest across a field from aerial imagery so that farmers may be alerted to problematic areas in a timely fashion to inform their management decisions. Meanwhile, the

release of these datasets will support numerous avenues of research for computer vision in remote sensing for agriculture.

- Chapter 4 addresses the challenge of limited labeled data in remote sensing and earth observation by introducing GenCo, a generative-based contrastive learning framework tailored for few-shot learning tasks. Focusing on RS-EO applications, the study overcomes the data scarcity issue through a two-stage training process involving contrastive-learning-based pre-training, along with an auxiliary generator. Ultimately, the pre-trained generator will enrich the information by fine-tuning it with minimal labeled data.

- Chapter 5 ventures into the realm of agricultural challenges, specifically focusing on N management, a critical aspect in meeting the growing global food demand amidst declining natural resources and climate change. Addressing the uncertainty in existing N management practices, the study proposes an innovative deep RL framework integrated with crop simulations. This approach, advancing over traditional methods, involves training N management policies with deep RL algorithms for maize crops in diverse locations like Iowa and Florida, offering a more scalable and comprehensive solution.

- Chapter 6 introduces an innovative crop management framework, employing deep RL alongside the Decision Support System for DSSAT for crop simulations. Uniquely, it incorporates LMs to transform crop and environmental data into descriptive narratives, significantly enhancing the RL agent's decision-making capabilities.

- Chapter 8 discusses the broader domain of related research, concludes the dissertation by offering final remarks and delves into discussing potential future directions.

1.2 Related Work

1.2.1 Contrastive Representation Learning

Unsupervised and self-supervised learning (SSL) methods have proven to be very successful for pre-training deep neural networks [37], [41]–[46]. Recently, methods like MoCo [2], [6], SimCLR [3], BYOL [1] and others such as [47]–[49] based on contrastive learning methods have achieved state-of-the-art performance. These approaches seek to learn by encouraging the attraction of different views of the same image ("positive pairs") as distinguished from "negative pairs" from different images [50]. Several approaches have sought to build on these base frameworks by making modifications that better incorporate the invariant properties and structure of the input data or task output. Specifically pertinent to the current work, [51] extended the SimCLR framework through the incorporation of a pixel-to-propagation module and additional pixel-level losses to improve performance on downstream tasks requiring dense pixel predictions.

1.2.2 Remote Sensing Datasets

Aerial images have been widely explored over the past few decades [10], [37], [52]–[55], but the datasets for image segmentation typically focus on routine, ordinary objects or street scenes [21]. Many prominent datasets including Inria Aerial Image [56], AV+[37], EuroSAT [57], and DeepGlobe Building [58] are built on low-resolution satellite (e.g. Sentinel-1, Sentinel-2, MODIS, Landsat) and only have limited resolutions that vary from 800 cm/pixel to 30 cm/pixel and can scale up to 5000×5000 pixels. Those datasets featuring segmentation tend to explore land-cover classification or change detection [59], [60].

Pertaining to aerial agricultural imagery, datasets tend to be either low-resolution (>10 m/pixel)

satellite [61], [62] or very high-resolution (<1 cm/pixel) imagery taken from UAV or on-board farming equipment [63], [64]. The Agriculture-Vision dataset [27], [65] introduced a large, high-resolution (10 cm/pixel) dataset for segmentation, bridging these two alternate paradigms.

1.2.3 Hallucination

Hallucination is initially proposed to solve the scarcity of data in the classification task [66]. Then, this idea is kept updated and applied in different areas [20], [67]–[73], such as object detection, aerial navigation, skeleton-based action recognition, and face generation. While image-level hallucination benefits few-shot recognition by synthesis of novel view [74] or introducing random noises [20], most of the work applies hallucination in the feature space. The work from [66] generates novel class features by transforming shared features in base classes. Authors of [69] build a hallucination framework in the region of interest feature space object to enhance the object detection performance. More recent work shows that this hallucination mechanism also benefits 3D human pose estimation by generating novel motion sequences [75]. While hallucination is effective in different learning tasks, to the best of our knowledge, the performance and application of hallucination in SSL are fully unexplored.

1.2.4 Feature-Level Augmentation

Hallucination relies on effective feature-level augmentation or manipulations. For instance, in [76], a task-agnostic feature augmentation approach is proposed to enrich the training data with minimal additional computation. The authors of [77] also explore similar ideas in the domain of few-shot learning. Building upon this, the concept is adapted to sentence representation learning [78], [79] and few-shot learning tasks in remote sensing [80]. More recently, [81] applied feature augmentation based on a

meta-learning technique.

1.2.5 Few-Shot Learning

Among various methods to speed up training and enhance label efficiency, such as neural network pruning [82]–[84], simulation-based training [85], [86], or few-shot learning. Few-shot learning is the most promising one in various application domains like healthcare [87], [88], remote sensing [33], learning tasks including classification [84], [89], object detection [39], semantic segmentation [90], and robot learning [91]. Generally, previous works can be roughly cast into three categories: metric-based, optimization-based, and hallucination-based.

The key idea of metric-based approaches is to learn good embeddings with appropriate kernels. Previous results from [92] propose applying a siamese neural network for few-shot classification. Following that, [31] presents a Relation Network by replacing the L1 distance between features with a convolutional neural network (CNN)-based classifier and updating the mean squared error (MSE) with cross-entropy; the triplet loss is utilized to improve the model's performance [93]. The results in [94] further add extra self-supervised tasks to enhance generalization capacity.

Optimization-based methods aim to learn through gradient backpropagation. Representative works include MAML [30], which realizes quick adaptation from good initialization, Reptile [95], which simplifies the learning process of MAML, and MetaOptNet [96], which incorporates the support vector machine (SVM) as a classifier.

Hallucination-based methods seek to learn generators to generate unseen samples. Works from [20], [71], [97] show that such a strategy of hallucination improves the test results and enhances the generation of models.

Few-Shot learning for RS-EO has received increased attention in recent years, [33]. While much of the work is focused on scene classification [34], [98]–[100], other recent approaches examine semantic segmentation tasks [101]–[103].

1.2.6 Pairing Supervised and Self-Supervised Learning

Some methods utilize the SSL loss as supplemental losses during the supervised training process [104], [105]. Often, additional efforts are needed to calibrate (i.e., re-weight) these losses when crossing different domains [106]. More straightforward and effective methods come from supervised fine-tuning [107]. While SSL encourages the learning of general-purpose features, the adaptation of features on the new task can be realized with only a few labeled samples. Within RS-EO, very recent work has looked to combine SSL and few-shot learning for scene classification [108] and segmentation [109].

1.2.7 LM for Decision-making

In recent years, there has been a surge in studies utilizing pre-trained LMs as decision-making agents. These models' remarkable capabilities have been harnessed across various domains, generating improved control plans for diverse robots and agents [110]–[120]. Notably, researchers of [121] developed LM-based agents for user interface (UI) interactions, while ReAct [122] integrated action decisions with natural language reasoning, demonstrating promising results.

1.2.8 Reinforcement Learning in Agricultural Management

Reinforcement learning, as a sub-field of machine learning, aims to solve SDM problems by letting an agent directly interact with the environment and learn from trial and error [123]. As a pioneering work,

[124] proposed to use a simple RL method and crop simulations to optimize the management of wheat crops in France. [125] studied the use of SARSA(λ), an on-policy RL method, and crop simulations to optimize the irrigation for the maize crop in Texas, US. However, the state and action spaces in [124], [125] were quite small due to the curse of dimensionality from which early RL methods suffered. For instance, the state space in [125] has only one state, i.e., total soil water (TSW) level. In contrast, modern RL methods, represented by deep RL, are able to handle extremely large state and action spaces due to the use of DNNs and have achieved remarkable or superhuman performance on a variety of high-dimensional problems such as gaming [126], [127], data center cooling [128], and robotic control [129]–[131].

Deep RL based on the proximal policy optimization (PPO) algorithm was used in [132] to optimize the fertilizer management for the wheat crop. Additionally, [133] studied the use of PPO to optimize the irrigation management for russet potatoes. However, the study is quite coarse and the results are not promising. For instance, in terms of results, [133] included only a simple learning curve showing the normalized reward, while the variables farmers mostly care such as yield and management cost were not included. Additionally, the trained policy performed much worse than a simple policy which applies a constant amount of water.

1.2.9 Crop Models

Crop models can simulate crop growth in response to soil, water, nutrient, and weather dynamics. They are playing increasingly important roles in the development of sustainable agricultural management because field and farm experiments require large amounts of resources and may still not provide sufficient information in space and time to identify appropriate and effective management practices [134]. The development of crop models dates back to 1950s. In the past seven decades, many crop models of

varying complexities have been developed by different groups, which include Agricultural Production Systems Simulator (APSIM), CERES (now contained in the DSSAT Suite of crop suite), CROPSYST, EPIC, WOFOST, and COUP. See the survey paper [134] and a comparison of different crop models for yield response prediction [135]. Among the existing crop models, the ones that are extensively used globally are APSIM and DSSAT, which are still constantly evolving and currently open-source to facilitate community-based development.

Most of the existing crop models need the management practices to be pre-specified before the start of a simulation, while RL-based training of management policies requires the management practices to be determined according to the soil, plan and weather conditions on a daily or weekly basis during the simulation. In light of this, the authors of [132] developed the CropGym environment for training of N management policies, which provides an interface to Open AI Gym [136], a widely used toolkit for RL research, and enables an RL agent to interact with the crop environment weekly. However, CropGym is based on the LINTUL-3 model [137] for the wheat crop, which has limited use. In a similar spirit, [133] presented another crop environment with the Open AI Gym interface for the russet potato based on the SIMPLE crop model [138], which, again, has limited use, potentially because the model is over-simplified. Recently, a Gym-DSSAT environment for the maize crop, which is based on the widely used DSSAT suite of crop models and provides a Gym interface, was developed [139] and enables an RL agent to interact with the environment on a daily basis. However, there have been no results on the use of Gym-DSSAT for training crop management policies up to now.

Chapter 2

Improving the Performance of Unsupervised Visual Representation Learning

2.1 Research Overview

Contrastive learning models based on Siamese structure have demonstrated remarkable performance in self-supervised learning. Such a success of contrastive learning relies on two conditions, a sufficient number of positive pairs and adequate variations between them. If the conditions are not met, these frameworks will lack semantic contrast and be fragile on overfitting. To address these two issues, we propose *Hallucinator* that could efficiently generate additional positive samples for further contrast. The Hallucinator is differentiable and creates new data in the feature space. Thus, it is optimized directly

with the pre-training task and introduces nearly negligible computation. Moreover, we reduce the mutual information of hallucinated pairs and smooth them through non-linear operations. This process helps avoid over-confident contrastive learning models during the training and achieves more transformation-invariant feature embeddings. Remarkably, we empirically prove that the proposed Hallucinator generalizes well to various contrastive learning models, including MoCoV1&V2, SimCLR and SimSiam. Under the linear classification protocol, a stable accuracy gain is achieved, ranging from 0.3% to 3.0% on CIFAR10&100, Tiny ImageNet, STL-10 and ImageNet. The improvement is also observed in transferring pre-train encoders to the downstream tasks, including object detection and segmentation.

2.2 Improved Representations

In this section, we first introduce the overall process of hallucination for contrastive representation learning in Section 2.2.1. Secondly, we highlight the center cropping method we used, which is crucial to effective hallucination or generation of new samples in Section 2.2.2. Then, we introduce the *Hallucinator* incorporated into our contrastive models in Section 2.2.3. Finally, we visualize and discuss the critical properties of the hallucination method from two perspectives in Section 2.2.4: the similarity of positive samples and the uniformity of the feature distribution.

2.2.1 The Overall Pipeline

Taking MoCo [6] as an example, we illustrate how a *Hallucinator* can be plugged into a contrastive learning model in Figure 2.1. Our architecture takes one image x as input. Then, the input x is augmented into two views x_1 and x_2. Each view will be processed by an encoder consisting of a backbone (e.g., ResNet) and a projector (e.g., an MLP head). After the encoders, output vectors q and k are obtained, forming one

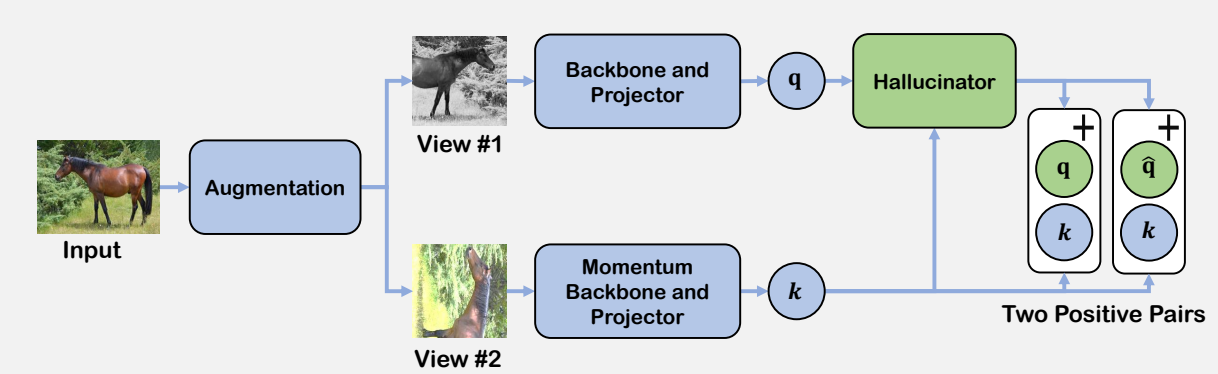

Figure 2.1: Illustration of contrastive learning (MoCoV2 [6]) with *Hallucinator*. The *Hallucinator* is added after the backbones and projector for feature-level manipulation. While *Hallucinator* feeds original feature vector q forward, it additionally provides hallucinated feature $\hat{q}$ for further contrast.

positive pair (q, k). Then, this positive pair (q, k) is fed to a *Hallucinator*. Notably, *Hallucinator* is only added to one branch of the framework to generate an additional positive feature $\hat{q}$. Together with feature vector k, $\hat{q}$ and k form as an extra positive pair $(\hat{q}, k)$ during the training. Based on different contrastive learning models, the loss functions keep intact, and the average loss of these two positive pairs is computed for back-propagation. The same paradigm could be applied to SimCLR, SiamSiam and other contrastive learning models. We illustrate further details about pipelines and loss functions of other models in this dissertation in the Supplementary Material (Section 1.1).

2.2.2 Center Cropping

In contrastive learning, data augmentations aim to ensure the performance of pre-trained representations invariant to nuisances. Among all these methods, random crop plays the most critical role in all the contrastive learning models. Generally, views (cropped tiles) generated by random cropping are diversified, successfully covering all the semantic information over the whole image. However, such a cropping method is likely to generate false positive patches [12]. In other words, patches randomly cropped from the original images do not necessarily share the overlapped pixels and sufficient common information. Therefore, these

false positive pairs may be fooling models during training, causing representations to be sub-optimal. Importantly, the issue will be exacerbated if we generate further hallucinated samples based on false positive pairs, which misleads the overall training beyond the sweet spot.

To tackle this issue, we first apply center cropping $\mathbb{C}_{crop}$ to the original image, getting a relatively smaller image $\hat{I}_{x,y}$. Then, random cropping $\mathbb{R}_{crop}$ is applied to $\hat{I}_{x,y}$. More specifically, the center cropping can be formulated as

$$\hat{I}_{x,y} = \mathbb{C}_{crop}(I_{x,y}, p), \tag{2.1}$$

where $I_{x,y}$ is the input image with (x, y) as the coordinate of the images' center. After center cropping, we keep the center of $\hat{I}_{x,y}$ unchanged. Meanwhile, with the original shape of $I_{x,y}$ defined as (h, w), we define the shape of cropped image $\hat{I}_{x,y}$ as $(\hat{h}, \hat{w})$. The p denotes a ratio of cropped length over the original length, i.e., $p = \frac{\hat{w}}{w} = \frac{\hat{h}}{h}$. Unless otherwise mentioned, we set $p = 0.5$ for all experiments in this dissertation.

While center cropping effectively avoids false positive pairs, it reduces the operable region for random cropping and generates positive views with a similar appearance. We, therefore, adopt center-suppressed sampling [12] with a sampling method following a beta distribution $\beta(\alpha, \alpha)$ (i.e., a U-shaped distribution). Concretely, $\beta(\alpha, \alpha)$ assigns a lower probability to the center of the $\hat{I}_{x,y}$ and gives greater probability to its boundary, increasing the variance between views x_1 and x_2. Together, we summarize the process to obtain these two views as

$$\begin{aligned} x_1 &= T(\mathbb{R}_{crop}(\hat{I}_{x,y}|\alpha)), \quad \text{s.t.} \quad \alpha < 1 \\ x_2 &= T(\mathbb{R}_{crop}(I_{x,y}|\alpha)), \quad \text{s.t.} \quad \alpha < 1, \end{aligned} \tag{2.2}$$

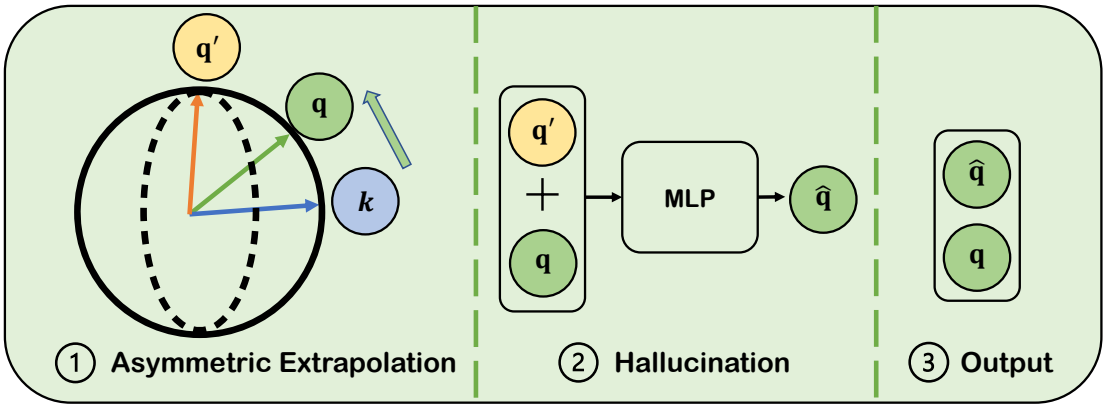

Figure 2.2: The *Hallucinator*. Stage 1: The original feature vector q is extrapolated to the opposite direction of feature vector k, forming $\hat{q}$ in a linear way. Stage 2: A non-linear transformation to smooth extrapolated features concatenated with q and q'. Stage 3: Output original q and hallucinated $\hat{q}$.

where T denotes data augmentations, including color jittering, random grayscale, Gaussian blur and horizontal flipping. Further, $\mathbb{R}_{crop}(...|\alpha)$ represents random cropping following the $\beta(\alpha, \alpha)$ distribution, and α is set to less than 1 to ensure an increasing sampling probability as the pixel's coordinates go beyond the center. A visualization of the center sampling method can be found in Supplementary Material (Section 1.2).

2.2.3 Hallucinator

Asymmetric Feature Extrapolation. The first objective of this module is to introduce an additional positive pair without introducing extra computations. Therefore, the feature-level operation is preferred compared to image-level operations. To achieve this, *Hallucinator* is plugged in after the views x_1 and x_2 are encoded. As a result, the hallucinated (generated) feature is purely based on the two feature vectors q and k.

Meanwhile, since harder positives improve pre-trained encoders' generalization capacity [5], the hallucinated features in positive pairs are favorable if they share less mutual information. Previous work

illustrates that a symmetric positive extrapolation is effective in generating hard examples for MoCo [14]. Concretely, two positive features are combined with weighted addition, pushing positive features apart. However, the hallucination process is asymmetric, i.e., *Hallucinator* is only added in one of the branches of the model. Then, we propose to apply the singe-side feature extrapolation to the feature vector q, as shown in the first stage of Figure 2.2. Additionally, we simplify the sampling strategy of weights for extrapolation from a beta distribution to a uniform distribution. To be specific, we summarize the positive extrapolation in our method as follows:

$$q' = (1 + \lambda)q - \lambda k \quad \text{s.t.} \quad \lambda \sim U(\beta_1, \beta_2), \tag{2.3}$$

where λ is sampled from a uniform distribution $U(\beta_1, \beta_2)$. The parameters β_1 and β_2, which define the boundary of the uniform distribution, are set to 0 and 0.1 by default.

Hallucination. Positive extrapolation is based on mixup [140], i.e., a linear transformation. While positive extrapolation has been proven beneficial to generating hard examples, this linear feature transformation might have a relatively limited capacity to synthesize new feature vectors. This assumption is based on the more satisfactory performance of the non-linear mixup over the original one [141]. Similarly, if non-linearity is introduced to the feature generation, the generated vector will benefit more from the training and boost the performance of downstream tasks.

To empower our model with the capacity of non-linear fitting, we introduce the hallucination process. Specifically, we first concatenate q' from the equation 2.3 and the feature vector q together. Then, we use the concatenated feature (q, q') as an input of a non-linear transformation function $H_\theta(.)$ that can be instantiated with n linear layers and a ReLU layer between two successive layers. When $n = 0$, $H_\theta(.)$ is a

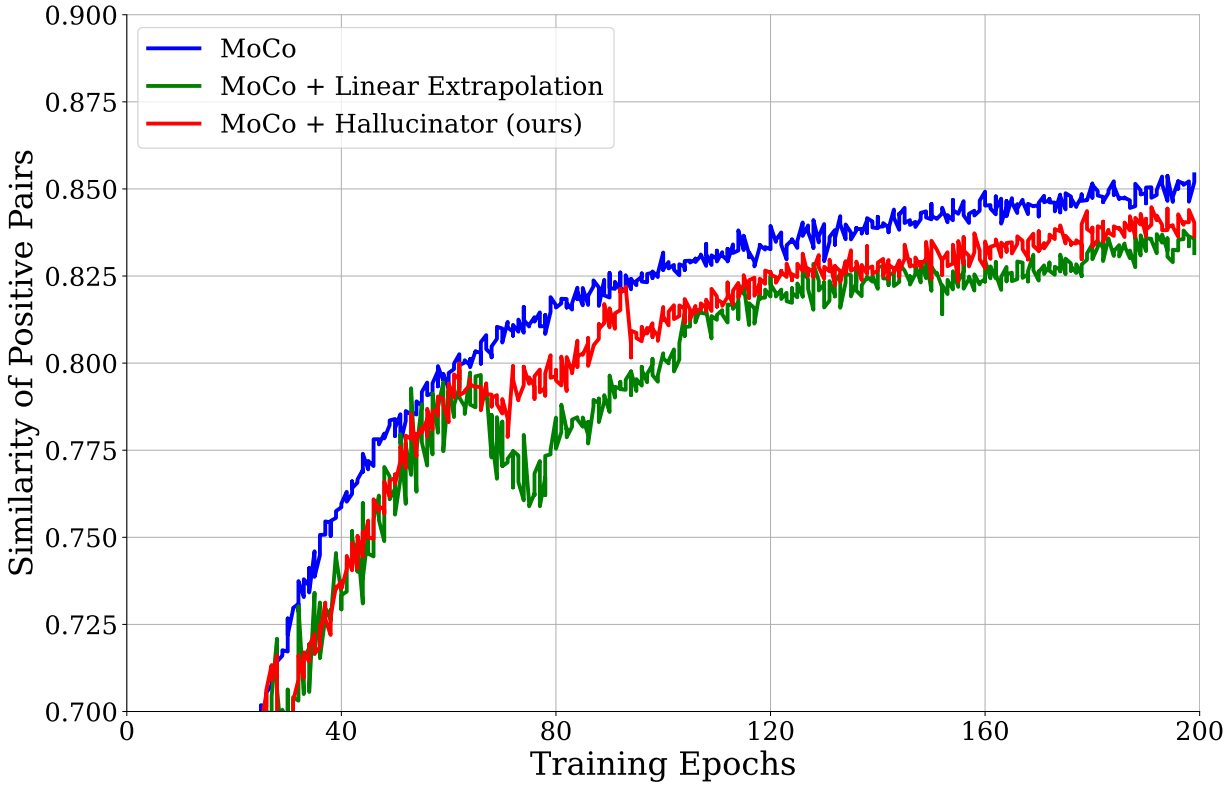

Figure 2.3: Similarity of positive pairs in training. Smaller values indicate less mutual information and better representation [5], [12], [14]. As *Hallucinator* incorporates non-linearity extrapolation, it guarantees smooth training and harder positive features.

non-parametric module, forming an identity function. Such a setting performance is relatively sub-optimal. Empirically, we find that $n = \{2, 3\}$ performs well as the *hallucinator* becomes non-linear and more powerful. We set $n = 3$ by default as its results are slightly better. With the transformation function $H_\theta(.)$, the hallucinated feature is defined as

$$\hat{q} = H_\theta(q, q'), \tag{2.4}$$

where θ is the parameters of $H_\theta(.)$. Notably, $H_\theta(.)$ is differentiable, allowing us to back-propagate the loss of contrastive learning. Therefore, we update not just the parameters of the encoders and projectors but also the parameters θ of the hallucinator. The second stage of Figure 2.2 illustrates the proposed

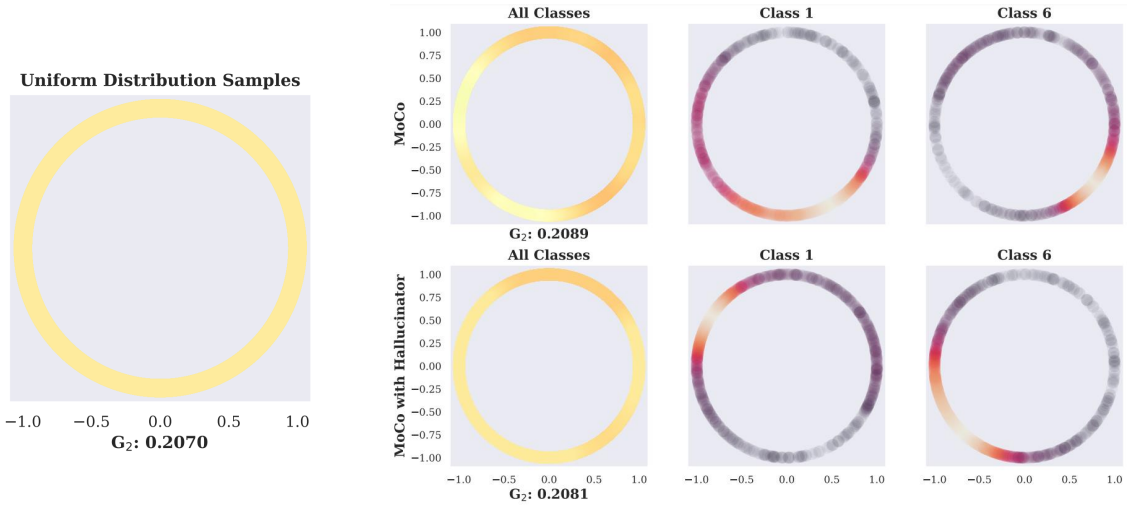

Figure 2.4: A measure of uniformity based on G_2 potential. We plot 10,000 feature vectors with Gaussian kernel density estimation (KDE) in $\mathbb{R}^2$. The left subplot illustrates the feature vectors from a uniform distribution. The three feature distributions on the right in the first row visualize the features from MoCoV2 [6]. The other three feature distributions in the second row demonstrate MoCoV2 with *Hallucinator*. *Hallucinator* benefits the uniformity with a smaller value of G_2.

hallucination process. Following that, we take q and $\hat{q}$ as the output of *Hallucinator*.

2.2.4 Discussion and Visualization

To better understand the behavior of *Hallucinator*, we discuss two critical properties that may contribute to and explain its effectiveness. For visualization, we train MoCoV2 [6] with a standard ResNet-18 [22] on Cifar-10 [142].

Similarity of Positive Pair. While hard positives share less mutual information, thus having a smaller cosine similarity value, the performance of downstream tasks will be enhanced [14]. Based on Figure 2.3, the proposed *Hallucinator* generates harder positives with smaller values of similarity. Consequently, it helps contrastive learning models obtain more nuisances-invariant features. More importantly, different from linear symmetric extrapolation, our training curve is relatively stable without many oscillations introduced. This observation indicates that *Hallucinator* is successfully optimized with the overall framework. Meanwhile, hallucinated features nicely fit into the contrastive learning task.

Uniformity. We continue to analyze the performance of the proposed model from the perspective of uniformity. Notably, feature vectors from contrastive learning should be roughly uniformly distributed on a unit hyper-sphere, which ensures maximal information is preserved in the feature space [143]. In other words, the closer the feature distribution is to the uniform distribution, the more the feature benefits downstream tasks. To quantize uniformity, we follow previous work [143] to compute the average value of the Gaussian potential kernel (i.e., Radial Basis Function kernel) of positive features:

$$G_t(q, k) \triangleq e^{-t\|q-k\|_2^2} \quad \text{s.t.} \quad t > 0, \tag{2.5}$$

where t is a fixed parameter set as 2 for all the experiments. We visualize the feature vectors by mapping them to two-dimension feature space and applying l_2 normalization in Figure 2.4. $G_2 = 0.2070$ for uniformly distributed samples, whereas features from MoCoV2 have $G_2 = 0.2089$. If we plugin *Hallucinator* into MoCo, *Hallucinator* provides further contrast during the training with extra positives introduced, giving more uniformly distributed features and a decreased G_2 value, i.e., 0.2081. Additionally, we visualize the features of two classes of Cifar-10. Each of these features is well-clustered. With better uniformity, clusters' overlapping decreases, forming more linearly separable features. Therefore, we could observe that the overlapping of feature clusters between class 1 and class 6 in MoCo is larger than the one with *Hallucinator* plugged in.

Dataset	CIFAR-10		CIFAR-100		Tiny ImageNet		STL-10	
Hallucinator	✗	✓	✗	✓	✗	✓	✗	✓
MoCoV1	88.31	**88.94**	60.94	**61.81**	44.65	**45.53**	88.19	**90.09**
MoCoV2	87.21	**89.23**	59.70	**61.26**	47.12	**47.95**	89.32	**90.46**
SimCLR	89.66	**90.11**	60.94	**61.43**	45.22	**46.30**	89.07	**89.98**
SimSiam	90.47	**90.78**	63.39	**64.38**	43.66	**44.96**	87.79	**88.16**

Table 2.1: Linear classification results for different contrastive methods and datasets in small scales.

Method	Backbone	Epoch	IN-200	IN-1K
MoCoV1	ResNet-50	100	62.19	57.27
MoCoV1(Ours)	ResNet-50	100	**63.46**	**59.17**
MoCoV2	ResNet-50	100	62.57	64.41
MoCoV2(Ours)	ResNet-50	100	**63.58**	**64.97**
SimCLR	ResNet-50	100	62.22	61.23
SimCLR(Ours)	ResNet-50	100	**63.02**	**61.71**
SimSiam	ResNet-50	100	62.80	63.11
SimSiam(Ours)	ResNet-50	100	**63.52**	**63.55**

Table 2.2: Linear classification results on IN-200 and IN-1K.

2.3 Experiments and Results

2.3.1 Linear Classification Protocol

Following the previous protocol, we first evaluate the proposed method by linear classification of frozen features. For each dataset, we report the top-1 classification accuracy on the validation set.

Results on Small-Scale Datasets. The classification results on Cifar10&100, Tiny ImageNet and STL-10 are reported in Table 2.1. With *Hallucinator* introduced, we notice a stable improvement over the baselines ranging from 0.31% to 1.98%. Notably, such improvements do not introduce extra computations and generalize well to various models.

Results on ImageNet. For the results of ImageNet, we report the results at two different scales. First, we evaluate its performance on standard IN-1K (i.e. ImageNet-1K), which consists of 1000 classes. Second, we test the proposed method in IN-200 (i.e. ImageNet-200) with 200 randomly selected classes. We report

Acc.(%)	Center Cropping	Asymmetric Extrapolation	Hallucination
62.57	✗	✗	✗
63.58(+1.01)	✓	✓	✓
62.58(+0.01)	✓		
62.82(+0.25)		✓	
62.84(+0.27)			✓
63.07(+0.50)		✓	✓

Table 2.3: Ablation of the different modules in the proposed method.

the corresponding results in Table 2.2. We found that *Hallucinator* essentially benefits MoCoV1 with 1.27% and 1.89% improvements for IN-200 and IN-1K accordingly. For MoCoV2, SimSiam and SimCLR, we notice a gain in accuracy ranging from 0.44% to 1.01%. On average, the gains are more salient in IN-200.

2.3.2 Ablation Studies: Contributions of Modules

We report the results with or without crucial modules introduced in Section 2.2. According to Table 2.3, center cropping shows a similar performance to the original cropping method, successfully covering major semantic information of images. However, it successfully avoids false positives in pre-training, which is critical for hallucination. Asymmetric extrapolation benefits the performance of representation learning with reduced mutual information. This observation is consistent with the results shown in the symmetric extrapolation [14]. If we combine asymmetric extrapolation and the hallucination method, the performance of the model could be further boosted. However, it is still sub-optimal because of possible false positives in cropping.

Method	1N-1k	VOC detection			COCO detection			COCO instance segmentation		
	Top-1	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Random init	-	33.8	60.2	57.27	26.4	44.0	27.8	29.3	46.9	30.8
Supervised	76.1	53.5	81.3	58.8	38.2	58.2	41.2	33.3	54.7	35.2
InfoMin [5]	70.1	57.6	82.7	64.6	39.0	58.5	42.0	34.1	55.2	36.3
MoCoV1 [2]	60.6	55.9	81.5	62.6	38.5	58.3	41.6	33.6	54.8	35.6
MoCoV1(Ours)	**63.8**	**56.7**	**81.8**	**63.2**	**38.9**	**58.5**	**41.9**	**33.8**	**55.2**	**36.0**
MoCoV2 [6]	67.5	57.0	82.4	63.6	39.0	58.6	41.9	34.2	55.4	36.2
MoCoV2(Ours)	**68.0**	**57.4**	**82.7**	**63.9**	**39.3**	**58.8**	**42.3**	**34.6**	**55.5**	**36.4**

Table 2.4: Fine-tuning results on object detection tasks on PASCAL VOC and COCO, and instance segmentation on COCO. All models are pre-trained for 200 epochs on ImageNet-1K.

2.3.3 Transferring Features

The primary goal of representation is to learn transferrable features. We evaluate the transferability of features from the proposed method following the previous protocol [2], [3], [6], [7]. Then, we compare the representation quality by transferring them to downstream tasks, including VOC [8] object detection and COCO [10] object detection and instance segmentation. Notably, we re-implement all these experiments using the same settings in MoCo's detectron2 codebase [144].

Object Detection on PASCAL VOC. Following the paradigm of previous work [2], we use Faster R-CNN[145] as the object detection method using R50-C4 as the detector [9]. We train the model end-to-end on the **trainval2007+2012** and evaluate its performance on **test2007**. As shown in Table 2.4, we observe a stable gain range from 0.3 to 0.8 under different metrics on MoCoV1 and MoCoV2.

Object Detection and Instance Segmentation on COCO. We continue to report the detection and segmentation results on COCO using Mask R-CNN [9]. Similarly, we use the R50-C4 as the backbone. The model is trained in an end-to-end way on **train2017**. Then, the model is evaluated on **val2017**. Again, the proposed method benefits the object detection and segmentation tasks with various metrics as demonstrated in Table 2.4.

Chapter 3

Extended Agriculture-Vision: An Extension of a Large Aerial Image Dataset for Agricultural Pattern Analysis

3.1 Research Overview

A key challenge for much of the machine learning work on remote sensing and earth observation data is the difficulty in acquiring large amounts of accurately labeled data. This is particularly true for semantic segmentation tasks, which are much less common in the remote sensing domain because of the

incredible difficulty in collecting precise, accurate, pixel-level annotations at scale. Recent efforts have addressed these challenges both through the creation of supervised datasets as well as the application of self-supervised methods. We continue these efforts on both fronts. First, we generate and release an improved version of the Agriculture-Vision dataset [27] to include raw, full-field imagery for greater experimental flexibility. Second, we extend this dataset with the release of 3600 large, high-resolution (10cm/pixel), full-field, red-green-blue and near-infrared images for pre-training. Third, we incorporate the Pixel-to-Propagation Module [51] originally built on the SimCLR framework into the framework of MoCo-V2 [6]. Finally, we demonstrate the usefulness of this data by benchmarking different contrastive learning approaches on both downstream classifications *and* semantic segmentation tasks. We explore both CNN and Swin Transformer [146] architectures within different frameworks based on MoCo-V2. Together, these approaches enable us to better detect key agricultural patterns of interest across a field from aerial imagery so that farmers may be alerted to problematic areas in a timely fashion to inform their management decisions. Furthermore, the release of these datasets will support numerous avenues of research for computer vision in remote sensing for agriculture.

3.2 Datasets Analysis and Generation

3.2.1 Agriculture-Vision Dataset

The original AV dataset [27] consists of 94,986 labeled high-resolution (10-20 cm/pixel) RGB and near-infrared (NIR) aerial imagery of farmland. Special cameras were mounted to fixed-wing aircraft and flown over the Midwestern United States during the 2017-2019 growing seasons, capturing predominantly corn and soybean fields. After annotation, 512×512 *tiles* were extracted from the full-field images and

Figure 3.1: Left: Full-field imagery (RGB-only) constructed from the AV dataset. A field of this size is approximately 15,000×15,000 pixels which can yield many smaller tiles. Right: Sample imagery and labels for the fine-grained segmentation task.

then pre-processed and scaled. While this pre-processing produces a uniformly curated dataset, it naturally discards important information about the original data.

To overcome this limitation, we obtained the original raw, full-field imagery. We are releasing this raw data as full-field images without any tiling, as it has been demonstrated to be beneficial to model performance [65]. A sample image is shown in Figure 3.1 (left). The original dataset can be recreated from this new dataset by extracting the tiles at the appropriate pixel coordinates provided in the data manifest.

Raw Data for Pre-training

We identified 1200 fields from the 2019-2020 growing seasons collected in the same manner as in Section 3.2.1. For each field, we selected three images, referred to as *flights*, taken at different times in the growing season, resulting in 3600 raw images available for pre-training. We elect to include data from 2020 even though it is not a part of the original supervised dataset because it is of high quality, similar in distribution to 2019, and we wish to encourage exploration around incorporating different source domains into modeling approaches as this is a very central problem to remote sensing data. We denote this raw imagery plus the original supervised dataset (in full-field format) as the "Extended Agriculture-Vision Dataset" (AV+). Now, it is publicly available. The statistics of AV+ compared with AV are demonstrated

in Table 3.1.

One characteristic of remote sensing is data revisiting: capturing images from the same locations multiple times. Through data revisiting, the temporal information can serve as an additional dimension of variation beyond the spatial information alone. In AV+, a typical revisit time ranges from seven days to six months, capturing a field at different points during the pre-planting, growing, and harvest seasons. We provide an example of a revisit in Figure 3.3.

Table 3.1: Statistics of Agriculture Vision [27] and Extended Agriculture Vision (the part of raw imagery). We provide information about the number of images, image size, pixel numbers, color channels and the ground sample resolution (GSD). "cls.", "seg." and "SSL" stand for classification, segmentation and self-supervised learning, respectively.

Dataset	# of Images	Tasks	Image Size	Channels	Resolution (GSD)	# of pixels
AV	94,986	Cls./Seg.	512 × 512	RGB, NIR	10/15/20 cm/px	22.6B
AV+	3600	SSL	15,000 × 15,000	RGB, NIR	10/15/20 cm/px	810.0B

Fine-grained segmentation tasks for high-resolution remote sensing data, particularly for agriculture, are often overlooked because of the difficulty in collecting sufficient amounts of annotated data [147], [148]. To explore the transferability of the AV+ dataset and SSL methods to a very challenging, data-limited, in-domain (i.e., same sensor and geography) task, we construct a densely annotated dataset.

We collected 68 flights from the 2020 growing season that were not included in AV+ for this task. From these flights, 184 tiles with shape 1500×1500 were selected and densely annotated with four classes: soil, weeds, crops, and unmanaged area (e.g., roads, trees, waterways, buildings); an "ignore" label was used to exclude pixels which may be unidentifiable due to image collection issues, shadows, or clouds. The annotations in this dataset are much more fine-grained than those in the AV+ dataset. For example, whereas the AV+ dataset identifies regions of high weed density as a "weed cluster", this dataset identifies each weed individually at the pixel level and also labels any crop or soil in those regions by their appropriate class. A sample image and annotation are shown in Figure 3.1 (right).

The fine-grained nature and small dataset size make this a very challenging segmentation task: very young crops often look like weeds; weeds growing among mature crops are only detectable through an interruption in the larger structure of the crop row; unmanaged areas often contain grasses and other biomass which closely resemble weeds but are not of concern to the grower, and classes are highly imbalanced.

3.3 Methodology for Benchmarks

In this section, we present multiple methods for pre-training a transferable representation on the AV+ dataset. These methods include MoCo-V2 [6], MoCo-V2 with a Pixel-to-Propagation Module (PPM) [51], the multi-head Temporal Contrast based on SeCo [26], and a combined Temporal Contrast model with PPM. We also explore different backbones based on ResNet [22] and the Swin Transformer architecture [146].

3.3.1 Momentum Contrast

MoCo-V2 is employed as the baseline module for the pre-training task. Unlike previous work focusing only on RGB channels [2], [6], [26], we include the information and learn representations from RGB and NIR channels. In each training step of MoCo, a given training example x is augmented into two separate views, query x^q and key x^k. An online network and a momentum-updated offline network map these two views into close embedding spaces $q = f_q(x^q)$ and $k^+ = f_k(x^k)$ accordingly; the query q should be far from the negative keys k^-, coming from a random subset of data samples different from x. Therefore, MoCo can be formulated as a form of dictionary lookup, in which $k+$ and $k-$ are the positive and negative keys. We define the instance-level loss $\mathcal{L}_{inst}$ with temperature parameter τ for scaling [149] and optimize the dictionary lookup with InfoNCE [150]:

$$\mathcal{L}_{inst} = -\log \frac{\exp(q \cdot k^{+}/\tau)}{\sum_{k^{-}} \exp(q \cdot k^{-}/\tau) + \exp(q \cdot k^{+}/\tau)}. \tag{3.1}$$

3.3.2 Momentum Contrast with Pixel-to-Propagation Module

Compared with classical datasets such as ImageNet [21], COCO [10], and LVIS [54] in the machine learning community, low-level semantic information from AV+ is more abundant, with regions of interest corresponding more closely to "patterns" (i.e. areas of weed clusters, nutrient deficiency, storm damage) and less to individual instances. Therefore, pre-training MoCo-V2 beyond image-level contrast should be beneficial to downstream pattern analysis tasks.

Pixel-to-Propagation Module. Researchers of [51] added a Pixel-Propagation-Module (PPM) to the SimCLR framework and achieved outstanding results on dense downstream tasks. In PPM, the feature of a pixel x_i is smoothed to q_i^s by feature propagation with all pixels $x_{\hat{i}}$ within the same image I following the equation:

$$q_i^s = \sum_{x_j \in I} S(x_i, x_{\hat{i}}) \cdot G(x_{\hat{i}}), \tag{3.2}$$

where $G(\cdot)$ is a transformation function instantiated by linear and ReLU layers. $S(\cdot, \cdot)$ is a similarity function defined as

$$S(x_i, x_{\hat{i}}) = (max(cos(x_i, x_{\hat{i}}), 0))^{\gamma} \tag{3.3}$$

with a hyper-parameter γ to control the sharpness of the function.

Extend Pixel-to-Propagation Module to MoCo. Notably, previous work based on SimCLR requires

a large batch size, which is not always achievable. To generalize the PPM and make the overall pre-training model efficient, we incorporate the pixel-level pretext tasks into basic MoCo-V2 models to learn dense feature representations. As demonstrated in Figure 3.2, we add two extra projectors for pixel-level pretask compared with MoCo-V2. The features from the backbones are kept as feature maps instead of vectors to ensure pixel-level contrast. To decide positive pairs of pixels for contrast, each feature map is first warped to the original image space. Then the distances between the pixel i and pixel j from each of the two feature maps are computed and normalized. Given a hyper-parameter τ (set as 0.7 by default), i and j are recognized as one positive pair if their distance is less than τ. Then, we can compute the similarity between two pixel-level feature vectors, i.e., smoothed q_i^s from PPM and k_j from the feature map for each positive pair of pixels i and j. Since two augmentation views both pass the two encoders, we use a loss in a symmetric form following [51]:

$$\mathcal{L}_{PixPro} = -cos(q_i^s, k_j) - cos(q_j^s, k_i). \tag{3.4}$$

During the training, the loss $\mathcal{L}_{PixPro}$ from the PPM is integrated with the instance-level loss as shown in the equation (3.5). These two complementary losses are balanced by a factor α, set to 0.4 in all the experiments (see Supplemental: Additional Results - Balance Factor):

$$\mathcal{L} = \alpha\mathcal{L}_{inst} + \mathcal{L}_{PixPro}. \tag{3.5}$$

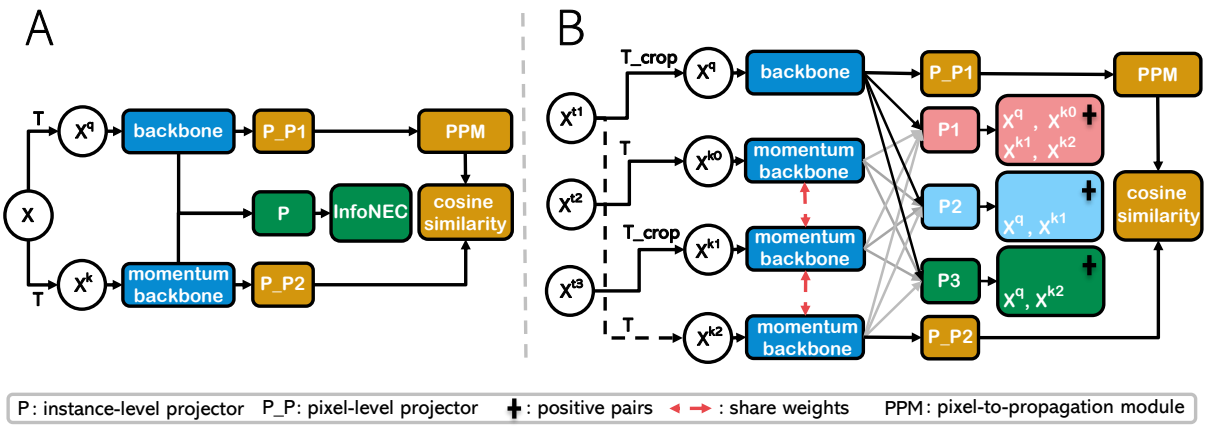

Figure 3.2: **A**. Diagram of MoCo-V2 with Pixel-to-Propagation Module (MoCo-PixPro). P includes a normally updated projector and a momentum-updated projector. For pixel-level pre-task, P_P1 is updated by gradient descent and P_P2 is the momentum projector. **B**. Diagram of Temporal Contrast with Pixel-to-Propagation Module (TemCo-PixPro). Query view x^q and key view x^{k_0} contain both artificial and temporal variance. Query view x^q and key view x^{k_1} contain only temporal variance. Query view x^q and key view x^{k_2} only contain artificial variance. Identical cropping T_{crop} is applied to x^{t_1} and x^{t_3}. Pixel-level contrast in only computed on x^q and x^{k_2}. For these two sub-plots, modules in navy blue ■ serve as encoders for feature extraction. Modules in brown ■ are designed for pixel contrast, which includes projectors, pixel propagation modules and the loss being used. Pink modules ■ represent instance-level contrast with embeddings space invariant to all kinds of augmentations. Similarly, modules in green ■ and sky blue ■ mean instance-level contrast but extract features invariant to artificial augmentation and temporal augmentation, respectively.

3.3.3 Temporal Contrast

While a pixel-level pretext task learns representations useful for spatial inference, we would like to learn a representation that takes advantage of the temporal information structure of AV+. Following the work of SeCo [26], additional embedding sub-spaces that are invariant to time are created. Since the backbones learn temporal-aware features through extra sub-spaces, it offers a more precise and general pattern analysis in downstream tasks. More specifically, we define a positive temporal pair by obtaining one pair of images from the same area (i.e., of the same field) but at different times, as shown in Figure 3.3. We explore whether the structure provided by the temporal alignment of positive temporal pairs provides more semantically significant content than naive artificial transformations (i.e., flipping, shifting) for pre-training.

Unlike in SeCo where images were separated with a constant time (3 months), the time difference between images from our data varies from 1 week to 6 months. We adapt SeCo as follows. First, we randomly select three tiles with 512×512 from the same field at identical locations but different times, which will be defined as x^{t_1}, x^{t_2} and x^{t_3}. Only random cropping T_{crop} is applied to the query image to generate the query view, i.e., $x^q = T_{crop}(x^{t_1})$. The first key view that contains both temporal and artificial variance is defined as $x^{k_0} = T(x^{t_1})$, where the T is the typical data augmentation pipeline used in MoCo. The second key contains only temporal augmentation compared with the query view. Therefore, we apply the exact same cropping window applied to the query image, $x^{k_1} = T_{crop}(x^{t_2})$. The third key contains only artificial augmentations, x^{k_2}=$T(x^{t_0})$. Following the MoCo and SeCo learning strategy [2], [26], these views can be mapped into three sub-spaces that are invariant to temporal augmentation, artificial augmentation and both variance. In this way, we fully explore the multi-time scale information in AV+ to improve the temporal sensitivity of encoders further. Since the temporal contrast does not necessarily cross seasons or enforce alignment of seasonality within a sub-space, we denote our approach as Temporal Contrast (TemCo).

3.3.4 Temporal Contrast with Pixel-to-Propagation Module

We create an integrated model (TemCo-PixPro) to capture the dense, spatiotemporal structure of AV+. Concretely, we merge PPM and TemCo into a single model to increase the encoders' spatial and temporal sensitivity.

To ensure efficient computation, we do not compute pixel-wise contrastive updates in each temporal sub-space. Instead, we assign two extra projectors for pixel-level contrastive learning. We include the PPM after the online backbone and one of the pixel-level projectors to smooth learned features. Then, we

Figure 3.3: Visualization of temporal contrast in AV+.

calculate the similarity of the smooth feature vectors and the momentum encoder features through a dot product. We illustrate the overall architecture of this model in Figure 3.2B.

3.4 Experimental Results

3.4.1 Downstream Classifications

Linear Probing. Following standard protocol, we freeze the pre-trained backbone network and train only a linear head for the downstream task. We train the models for 50 epochs using Adam optimizer with an initial learning rate of 0.0001 and report the top-1 classification validation set.

Figure 3.4 shows the impact of different weight initialization and percentages of labeled data in the downstream task. Consistent with previous research [26], there is a gap between remote sensing and natural image domains: ImageNet weights are not always an optimal choice in this domain.

MoCo-PixPro obtains the highest accuracy for the ResNet-18 backbone. As we compare the results of

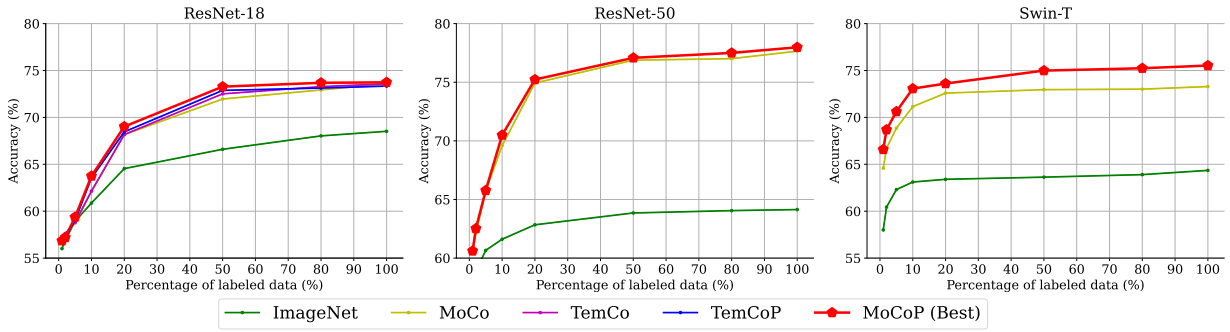

Figure 3.4: Accuracy under the linear probing protocol on AV+ classification. Results are shown from different pre-training approaches with different backbones.

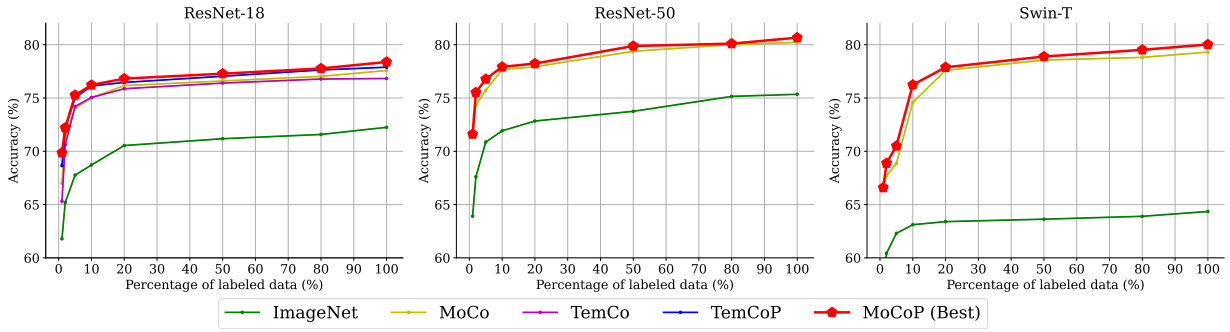

Figure 3.5: Accuracy under the non-linear probing protocol on AV+ classification. Results are shown from different pre-training approaches with different backbones, ResNet-18 (left), ResNet-50 (middle), and Swin-T (right), under different percentages of labeled data for the downstream task.

ResNet50 and Swin-T with fully labeled data, all Swin-T models underperformed their CNN counterparts.

Non-Linear Probing. We evaluate the frozen representations with non-linear probing: a multi-layer perceptron (MLP) head is trained as the classifier for 100 epochs with Adam optimization.

Classification results on AV+ classification under non-linear probing are shown in Figure 3.5. Consistent with results in the natural image domain [151], non-linear probing results surpass linear probing. Our SSL weights exceed ImageNet's weights by over 5% regardless of the amount of downstream data or backbone type. From the results of ResNet-18, the optimal accuracy between different pre-training strategies comes from either MoCo-PixPro or TemCo-PixPro, which is different from linear probing. Overall, MoCo-PixPro performs better than the basic MoCo model across different backbones.

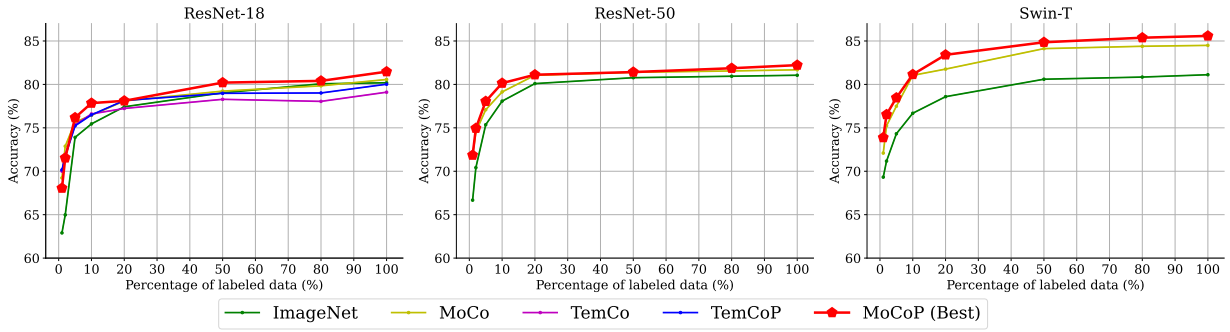

Figure 3.6: The accuracy under the end-to-end classification protocol on AV+. Results cover different pre-training approaches and backbones, varying from ResNet-18 (left), ResNet-50 (middle), and Swin-T (right). We also report the model's performance, tuned with different percentages of the fully labeled dataset, ranging from one percent to a hundred percent.

Fine-Tuning. Finally, we examine end-to-end fine-tuning with different percentages of labeled AV+ data for classification. We use the same architecture, learning schedule and optimizer as non-linear probing.

Our SSL weights show outstanding results in the low-data regions (<10% of data). For ResNet-18, MoCo-PixPro is better than the other models in all cases, whereas other SSL models demonstrate similar performance to ImageNet when labeled data is abundant. As we increase the backbone size to ResNet-50, our MoCo and MoCo-PixPro stably outperform ImageNet's model across all amounts of data, suggesting a greater capacity to learn domain-relevant features.

In Figure 3.6 (right), all models perform agreeably well in the Swin-T framework compared with weights from ImageNet. While fine-tuning was performed in the same manner as the ResNet models for fair comparisons, Swin-T shows the most promising performance in this end-to-end setting.

3.4.2 Semantic Segmentation on Extended Agriculture-Vision

As results are shown in Table 3.2, MoCo-PixPro performs the best for the ResNet-18 backbone when the encoder remains fixed during supervised training; this result is similar to that seen for classification. This result supports our hypothesis that AV+ has abundant low-level semantic information and including pixel-

Table 3.2: Results of Downstream Segmentation Task on AV+ using mean-IoU metric

Pretrained Weights	Backbone	mIoU (%) Fixed 1%	mIoU (%) Fixed 100%	mIoU (%) Fine-Tuned 1%	mIoU (%) Fine-Tuned 100%
Random	ResNet-18	18.89	21.37	19.02	26.94
ImageNet	ResNet-18	19.02	23.39	19.73	29.23
MoCo-V2	ResNet-18	22.36	27.83	**22.53**	**31.80**
MoCo-PixPro	ResNet-18	**23.71**	**30.60**	20.04	30.56
TemCo	ResNet-18	23.71	26.85	21.09	31.76
TemCo-PixPro	ResNet-18	22.97	28.60	21.32	31.66
Random	ResNet-50	19.42	21.82	18.71	26.37
ImageNet	ResNet-50	21.21	25.94	20.31	30.52
MoCo-V2	ResNet-50	24.25	31.03	**21.47**	**31.87**
MoCo-PixPro	ResNet-50	**25.76**	**32.35**	21.36	31.58
Random	Swin-T	15.89	20.10	22.68	37.14
ImageNet	Swin-T	20.00	22.40	30.96	43.01
MoCo-V2	Swin-T	25.51	30.60	28.12	41.02
MoCo-PixPro	Swin-T	**27.61**	**32.96**	**32.06**	**43.33**

level pre-task is critical for downstream learning tasks. When the encoder is unfrozen during supervised training, the basic MoCo-V2 shows the best results, but is not significantly better than TemCo or TemCo-PixPro. By scaling from ResNet-18 to ResNet-50, MoCo-PixPro outperforms ImageNet, especially when the encoder remains fixed. Importantly, unlike the ResNet-based models, the Swin Transformer-based MoCo-PixPro shows the best results across all variations in the setting. Another important observation is that the PPM benefits more as we scale up the models from ResNet-18 to ResNet-50 and then Swin-T. As the training epochs are all the same for all the pre-training, smaller backbones like ResNet-18 are more likely to get overfitted. When trained with ResNet-50, the performance drop of MoCo-PixPro is very small compared with MoCo. As we move to Swin-T, MoCo-PixPro eventually shows the best performance over other methods.

3.4.3 Comparison with Agriculture-Vision Results

The AV dataset was benchmarked on a downstream segmentation task with architectures based on the DeepLabV3 [152] framework. Since the previous results report mean Intersecion-over-Union (mIoU)

for 8 agricultural patterns, we re-trained our models using a U-Net architecture [153] including one more pattern, i.e., the storm damage. With a lightweight U-Net, smaller backbone, and much less training, our SwinT-based model outperforms the best results from [27] in the Table 3.3, demonstrating the effectiveness of our approach. Additionally, we demonstrate the effectiveness of pre-training and fine-tuning this multi-spectral data. AV and AV+ are beyond most conventional images, consisting of NIR-Red-Green-Blue (NRGB) channels. Therefore, we investigate the differences in semantic segmentation performance from multi-spectral images, including regular RGB and RGBN images. According to Table 3.3, NIR channels benefit the segmentation results over different backbones and segmentation methods.

For this eight-class segmentation task, we train the nine-class models using an Adam optimization with an initial learning rate of 0.01 and the one-cycle policy [154] for the learning rate adjustment. For fair comparisons and to be consistent with the supervised learning settings in [27], we use a batch size of 40 and 25,000 iterations with warmup training for 1,000 iterations.

Table 3.3: Comparison of mIoUs between the Agriculture-Vision model and our proposed U-Net-based model on Agriculture-Vision validation set.

Methods	Pre-trained Weights	Backbone	Channels	mIOU(%)	# Parameters
FPN[27]	ImageNet	ResNet-101	RGB	40.48	45.10M
U-Net	MoCo-V2	Swin-T	RGB	44.77	32.40M
U-Net	MoCo-PixPro	Swin-T	RGB	**45.92**	32.40M
FPN[27]	ImageNet	ResNet-101	RGBN	43.40	45.11M
U-Net	MoCo-V2	Swin-T	RGBN	46.15	32.40M
U-Net	MoCo-PixPro	Swin-T	RGBN	**48.75**	32.40M

3.4.4 Fine-Grained Semantic Segmentation

Unlike AV+, this dataset is severely limited by the availability of fine-grained segmentation labels. There are 184 tiles (from 68 flights) in this dataset that are split into training (70%), validation (15%), and test (15%). Again, we use a U-Net architecture with a ResNet-18 encoder. For training, we use a

Table 3.4: IoU for each model in the fine-grained semantic segmentation task considering different encoder weight initialization, architectures, and weight fixing schemes.

Weights	Architecture	IoU (Fixed-Weights)	IoU (Fine-Tuned)
Random	ResNet-18	39.05	42.19
ImageNet	ResNet-18	40.81	**45.47**
MoCo-V2	ResNet-18	**44.05**	43.97
MoCo-PixPro	ResNet-18	42.03	44.62
TemCo	ResNet-18	42.30	44.48
TemCo-PixPro	ResNet-18	43.45	43.91
MoCo-V2	ResNet-50	40.03	40.54
MoCo-V2	Swin-T	40.25	40.00
MoCo-PixPro	Swin-T	37.56	40.67
TemCo.	Swin-T	39.52	40.26

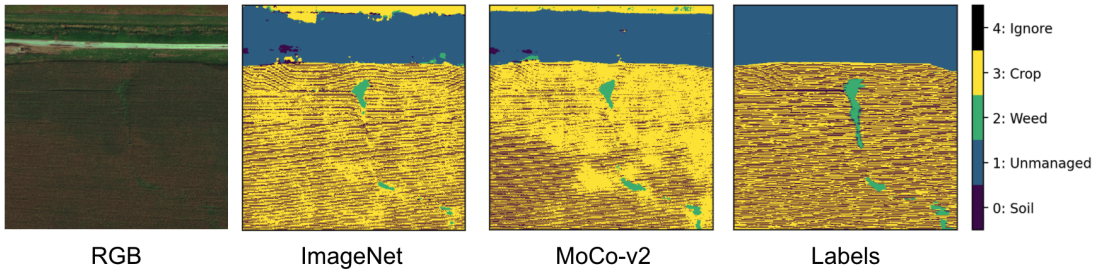

Figure 3.7: A sample output on the fine-grained segmentation task using fixed-encoder weights from ImageNet and MoCo-V2. The segmentation outputs are compared with both the original RGB image and the segmentation labels.

multi-class focal loss [155] to account for the strong class imbalance.

Results are shown in Table 3.4, and sample output is shown in Figure 3.7. Results improve across the board when both the encoder and decoder are fine-tuned. Although less dramatic than the results seen on the AV+ classification and segmentation tasks, some improvement over ImageNet weights is seen using the MoCo-v2 framework with ResNet-18 backbone for fixed weights. As seen on the other tasks, when the entire network undergoes fine-tuning, the ImageNet and SSL weights, specifically MoCo-PixPro, produce roughly the same performance on the downstream task. Additional per-class analysis is provided in the Supplemental. The ResNet-50 and Swin-T models performed relatively worse compared to the ResNet-18 models, which is unsurprising given the extremely small size of this dataset.

3.4.5 Land-Cover Classification on EuroSAT

We further prove that pretraining on the AV+ dataset benefits the downstream task in the broader remote sensing community. We conduct downstream classification experiments on EuroSAT [57]. EuroSAT addresses the classification challenge of land use and land cover with images from Sentinel-2. It consists of 27,000 labeled images and 10 classes over 34 European countries. We use the splits protocol of train/val following the work of [156].

We freeze the pre-trained backbones and add a linear layer to evaluate the learned representation in this classification task. The linear layer is tuned with 100 epochs using the Adam optimizer. The initial learning rate is set to 0.001 and is divided by 10 at the 60th and 80th epochs.

The results shown in the Table 3.5 compare the weights pre-trained from AV+ against other baselines. We notice that MoCo-V2 and our proposed MoCo-PixPro achieve 1.21% and 6.25% higher accuracy compared with ImageNet's weights accordingly. These results confirm not only the effectiveness of pre-training on AV+ but also AV+'s significant potential to generalize to the broader remote sensing field.

Table 3.5: Accuracy of the EuroSAT land-cover classification task using ResNet-18

Weights	Random	ImageNet	MoCo-V2	MoCo-PixPro
Accuracy (%)	63.21	86.32	87.53	**89.97**

3.4.6 Ablation Study: Number of Flights

We use a ResNet-18 backbone and basic MoCo-V2 for experiments. When the number of flights used for SSL is increased from 300 to 3600, we observe stable improvement in the downstream classification task under the non-linear probing setting; this gain is confirmed regardless of the fraction of the labeled dataset for tuning (see 'Supplemental: Additional Results' for more detailed exposition).

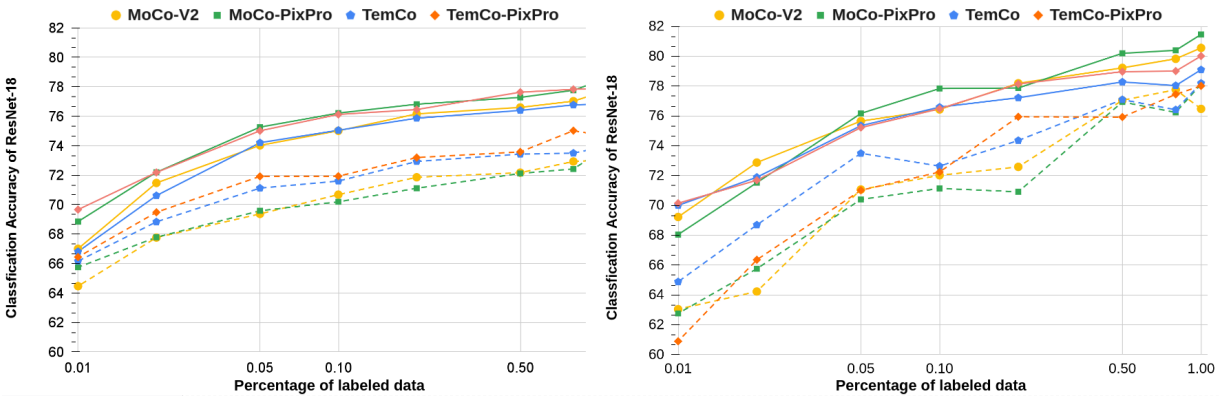

Figure 3.8: Ablation study on the pre-training size of data on different pre-training methods on two downstream tasks. Solid lines represent accuracy from 3600 flights while dashed lines represent accuracy from 1200 flights. Left: results from non-linear probing on downstream classification. Right: fine-tuning results on entire networks for downstream AV+ segmentation.

This improvement is seen for all examined SSL methods Figure 3.8 when the raw dataset is increased from 1200 to 3600 flights and evaluated under non-linear probing for classification and full-network fine-tuning for AV+ segmentation. Our SSL models' performance steadily grows as raw data size increases, suggesting that even more data may lead to even greater performance.

Chapter 4

GenCo: An Auxiliary Generator from Contrastive Learning for Enhanced Few-Shot Learning in Remote Sensing

4.1 Research Overview

Classifying and segmenting patterns from a limited number of examples is a significant challenge in remote sensing and earth observation due to the difficulty in acquiring accurately labeled data in large quantities. Previous studies have shown that meta-learning, which involves episodic training on query and support sets, is a promising approach. However, there has been little attention paid to direct fine-tuning techniques. This dissertation repurposes contrastive learning as a pre-training method for few-shot learning for classification and semantic segmentation tasks. Specifically, we introduce a generator-based contrastive

learning framework (GenCo) that pre-trains backbones and simultaneously explores variants of feature samples. In fine-tuning, the auxiliary generator can be used to enrich limited labeled data samples in feature space. We demonstrate the effectiveness of our method in improving few-shot learning performance on two key remote sensing datasets: Agriculture-Vision and EuroSAT. Empirically, our approach outperforms purely supervised training on the nearly 95,000 images in Agriculture-Vision for both classification and semantic segmentation tasks. Similarly, the proposed few-shot method achieves better results on the land-cover classification task on EuroSAT compared to the results obtained from fully supervised model training on the dataset.

4.2 Methodology

In this section, we first introduce the overall pipeline of pre-training and downstream few-shot learning in Section 4.2.1. Following this, we describe the proposed GenCo in Section 4.2.2. Lastly, we demonstrate how the generator from the contrastive learning model helps the downstream few-shot task by generating extra samples in feature space 4.2.3.

4.2.1 The Pipeline of Learning

In the first stage, we pre-trained different backbones with contrastive learning models on unlabeled data, as shown in Figure 4.1. To be more specific, there are roughly 1,300,000 images with shapes 512 $\times$ 512. These images are all randomly cropped from the raw images from AV+. We use all these unlabeled images as input to pre-train backbones without any supervision. After the pertaining, we move to the second stage, which fine-tunes the pre-trained backbones for the downstream few-shot tasks. Since the proposed contrastive learning is unsupervised, there is no information on any base classes, unlike the usual

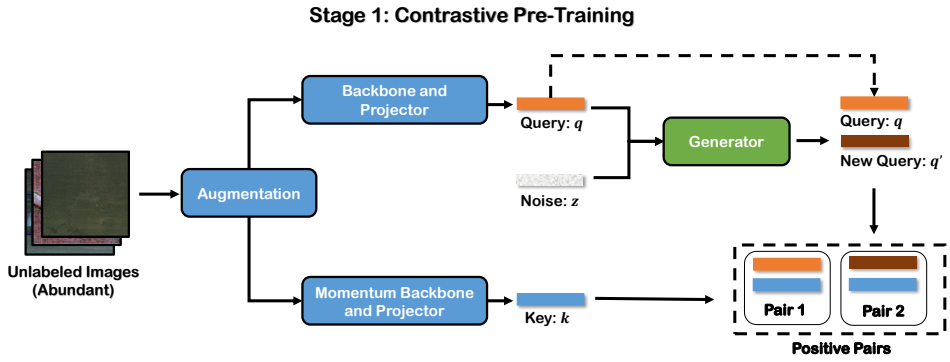

Figure 4.1: Illustration of the pre-training stage of GenCo. In the pre-training stage, we follow the training strategy of contrastive learning to train the backbones and projectors jointly with abundant unlabeled images. An additional feature vector q' is added by passing the real feature vector q and a noise vector z to the generator. The generator is trained end-to-end along with the contrastive learning framework.

settings for few-shot learning. Only k samples are provided for fine-tuning during the evaluation, where k varies from 1-10. The evaluation will include both base and novel classes based on different downstream tasks to optimize the classification accuracy or mIoU of agricultural patterns on Agriculture-Vision and land covers on EuroSAT.

4.2.2 Pre-training with Generator-based Contrastive Learning

Basic Framework. The framework of GenCo is shown in Figure 4.1. Specifically, GenCo can be trained with natural scene images that contain information about red, green, and blue channels similar to previous papers [3], [6], [7]. However, AV+ has extra information in the NIR channel. To fully explore knowledge from the pre-training dataset, we further add one channel to the backbones following the work of from [37].

In every training iteration, a training sample x is augmented into two different views named query x^q and key x^k. These views contain the same semantic meaning but also variations introduced from data augmentations, including spatial and color transforms. With an online network and a momentum-updated offline network proposed in [2], the training encourages these two views to be mapped into two similar

embedding spaces, i.e., q, k, as a positive pair. For feature vectors that are not encoded from x, we define them as k^-.

Feature Generation. Sufficient positive pairs in feature space help the performance of contrastive learning [3]. However, the large number of positive features relies on large batch sizes that may not always be accessible. To address this, we design a generator G that takes the query feature q and random noise z as input. Entries of z are assumed to follow a normal distribution with mean 0 and variance 0.1. We then generate a new sample q' in the feature space where $q' = G(q, z|\theta)$. Notably, G is a lightweight module with parameters θ. It is instantiated with three linear layers and a ReLU layer between two successive layers.

With this simple design, we eventually obtain an additional positive pair $p(q', k)$. Along with the original positive $p(q, k)$ one, the framework provided enhanced contrast and improve the quality of embeddings during the training with an additional little computation. Together, based on positive and negative pairs and a temperature parameter τ for scaling, the training loss function, i.e., InfoNCE [150], is then defined as follows:

$$\mathcal{L} = -\log \frac{\exp(q \cdot k/\tau) + \exp(q' \cdot k/\tau)}{\sum_{k^-} \exp(q \cdot k^-/\tau) + \exp(q \cdot k/\tau) + \exp(q' \cdot k/\tau)}. \tag{4.1}$$

4.2.3 Downstream Few-Shot Learning with Generator

Given the strong data augmentations in contrastive learning, the backbone features encoded can capture the key representation features. It should adapt to different data classes and types of downstream tasks

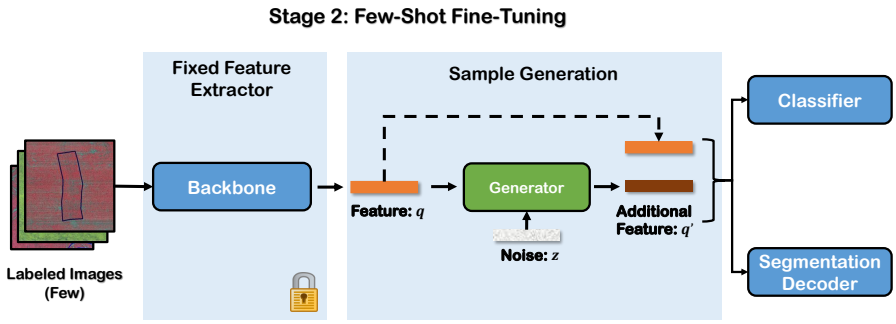

Figure 4.2: Illustration of the fine-tuning stage. In the fine-tuning stage with limited labeled images, we adapt the pre-trained backbone from stage 1 for feature extraction. The generator is brought from the GenCo to generate additional labeled data. During the tuning, the feature extractors are fixed, and fine-tuning is only involved in the generator, classifier and segmentation decoder.

without much learning effort. Therefore, one of the critical steps of the proposed method is to separate representation learning and downstream task learning into two stages.

With contrastive learning applied in the first stage, we freeze the encoder and fine-tune models with a few labeled images. We first create a small balanced training set with K images per class, i.e., K shots. These classes can either be seen or be novel classes. Then, each image is encoded into feature space, forming a vector q with label y. While the traditional fine-tuning method applied the feature q directly to downstream tasks, we take advantage of the generator G from the contrastive learning framework to generate labeled samples following the equation $q' = G(q, z|\theta)$, as shown in Figure 4.2. The q' shares the same label y with q, forming an extra label data (q', y). In other words, we eventually obtained 2K labeled data points given K images for each class, largely enriching the information for downstream learning tasks. During the fine-tuning stage, the generator G is differentiable and optimized simultaneously with the fine-tuning stage.

In the few-shot classification task, we add one fully connected layer to the backbones without introducing extra non-linearity. We assign randomly initialized weights to the added classification layer. In the few-shot

semantic segmentation task, we choose the lightweight segmentation model U-Net [153] for fine-tuning given limited training samples. Concretely, we add a five-layer decoder based on the encoders pre-trained from contrastive learning. In each layer of the decoder, it performs up-sampling by a 2×2 deconvolution layer to recover the original image sizes. While the size of the feature images increases, the number of channels reduces by half after each up-sampling. Empirically, this asymmetric encoder-decoder shows exceptional performance in the segmentation task on Agriculture-Vision.

4.3 Experiments and Results

In this study, we conduct various experiments to prove the effectiveness of our proposed methods. In Section 4.3.1, we introduce the used datasets in this dissertation. Then, we illustrate the evaluation metrics in Section 4.3.2. In the following Section 4.3.3, we demonstrate the necessary details to reproduce the experiments. Lastly, in Section 4.3.4, we report all the results based on the proposed metrics.

4.3.1 Datasets

Agriculture-Vision

Agriculture-Vision (AV) is a large aerial image database for agricultural pattern analysis. It contains 94,986 high-quality images over 3432 farmlands across the US. Totally, there are nine classes selected in the dataset under the advisement of agronomists, which include double plant, dry-down, endrow, nutrient deficiency, planter skip, storm damage, water, waterway, and weed cluster. With extreme label imbalance across categories, it is a challenge to train well-performing models for classification and segmentation tasks [157]. The original dataset is designed for semantic segmentation; we also create a "classification" version of the dataset by assigning a positive label if any presence of that class is included in the tile.

Extended Agriculture-Vision

For contrastive pre-training, we use the large-scale remote-sensing dataset, Extended Agriculture-Vision [37]. While the original Agriculture-Vision dataset contained only 512x512 tiles with semantic segmentation labels for agricultural patterns such as waterways, weeds, nutrient deficiency, etc., AV+ includes several thousand additional raw full-field images (upwards of 10,000 x 10,000 in dimension). Images consist of RGB and Near-infrared (NIR) channels with resolutions as high as 10 cm per pixel. As it also covers data that varies from 2017 to 2020, encoders pre-trained on this dataset should capture remote sensing, agriculture, and temporal features. Therefore, the embeddings pre-trained on Extended Agriculture-Vision should be adapted well to diverse downstream tasks such as agricultural pattern recognition and land-cover classification.

EuroSAT

EuroSAT is a dataset for the classification task of land use and land cover. All the satellite images are collected from Sentinel-2, covering 34 countries. There are 27,000 images in total, with ten types of labels corresponding to different land use cases. The class labels are evenly distributed, with each category consisting of 2,000 to 3,000 images. We use the split method for training and evaluation following the work from [26], [37], [57]. While there is a total of 13 channels in total, in this work, we focus on the RGB channels since it is a more general modality.

4.3.2 Experiment Setup and Evaluation Metrics

We evaluate the proposed method from two perspectives, i.e., the quality of embeddings and the required amount of labeled data for model adaptation. First, in the few-shot classification task, we compare

the performance of features from the backbone pre-trained on ImageNet, and the backbone pre-trained on AV+ using contrastive learning models. Similarly, in the few-shot semantic segmentation task, our embeddings pre-trained on AV+ and embeddings from COCO's weights are compared [10]. Second, to illustrate the learning efficiency of our method, we compare the two models' performances, one of which is fine-tuned on a few samples, and the other is trained in a supervised way with complete labeled data.

4.3.3 Implementations Details

Firstly, we introduce implementation details of the generator-based contrastive learning model to pre-train backbones, as shown in the left part of Figure 4.1 in Section 4.3.3. Then, we demonstrate details of few-shot experiments on classification and semantic segmentation tasks on Agriculture-Vision shown in the right part of Figure 4.2 in Section 4.3.3. Following this, we list the key parameters for evaluation in the few-shot semantic segmentation task in Section 4.3.3. Lastly, we report the necessary details for EuroSAT in Section 4.3.3.

Pre-training on Extended Agriculture-Vision

GenCo uses different ResNet as its encoder and two layers of MLP as a projector as the basis for our contrastive learning framework. The following generator G is instantiated with three linear layers and a ReLU layer between two successive layers. The output feature vectors, i.e., q, q', and k, have dimensions of 128, and there are 16,384 negative keys stored in the memory bank. We pre-train the generator-based contrastive learning model for 200 epochs using an SGD optimizer, a learning rate of 0.3, and a weight decay of 0.0001. The learning rate is adjusted to 0.03 and 0.003 at the 140th and 160th epochs accordingly. As this data is hyperspectral, we add one more channel to the encoder during the training for the NIR

input, and this extra channel is initialized with the same weights as the red channel.

Besides the proposed GenCo, we also conduct experiments using MoCo-V2 [6], SimSiam [7] and SimCLR [3] as alternative methods for pre-training. Results are also reported in the following sections. For MoCo-V2, we train the model with the exact same hyperparameters as the proposed method, given the similarity of the structure. For SimSiam, we train the models for 200 epochs with a batch size of 256. The learning rate = 0.05 with an SGD optimizer is used. The weight decay is 0.0001 and the SGD momentum is 0.9. For pre-training of SimCLR, we reproduce SimCLR with a smaller batch size of 512 and cosine appealed learning rate of 0.05. We follow the self-supervised learning paradigm in [158] without distillation for a fair comparison of the other two contrastive learning methods. All the experiments are conducted on a server with 8 GPUs.

Few-Shot Classification on Agriculture-Vision

The first set of experiments focuses on the classification formulation of the Agriculture-Vision task. We use ResNet-18, ResNet-50, and ResNet-101 as the backbones for fine-tuning. All backbones are fixed except the last fully connected layer and the generator G, which are learnable. Different from the optimization methods used in [2], we use Adam as an optimizer for all experiments with an initial learning rate set to 0.001. We train the classification models for 100 epochs with a batch size of 64.

Few-Shot Segmentation on Agriculture-Vision

Following the work from [27], we ignore storm damage annotations when performing evaluations due to their extreme scarcity. Similar to the fine-tuning strategy we used in the classification task, we freeze all backbones during training but with a learnable five-layer decoder. The decoders are randomly initialized and attached to the encoders, forming a lightweight and imbalanced U-Net. We use the AdamW optimizer

Table 4.1: Comparison of fine-tuning results between weights from supervised ImageNet and weights from GenCo on AV+ for the 10-shot classification

Pre-trained Weight	Backbone	Accuracy(%)
Sup. on ImageNet	ResNet-18	55.22 ±0.34
SimSiam on AV+	ResNet-18	63.12 ±0.71
SimCLR on AV+	ResNet-18	63.59 ±0.83
MoCoV2 on AV+	ResNet-18	64.17 ±0.62
GenCo on AV+	ResNet-18	**65.51** ±0.68
ImageNet	ResNet-50	56.53 ±0.43
SimSiam on AV+	ResNet-50	62.41 ±0.78
SimCLR on AV+	ResNet-50	63.55 ±0.77
MoCoV2 on AV+	ResNet-50	63.80 ±0.72
GenCo on AV+	ResNet-50	**64.82** ±0.70
Sup. on ImageNet	ResNet-101	54.34 ±0.45
SimSiam on AV+	ResNet-101	62.27 ±0.89
SimCLR on AV+	ResNet-101	63.50 ±0.80
MoCoV2 on AV+	ResNet-101	64.12 ±0.78
GenCo on AV+	ResNet-101	**64.62** ±0.49

with the learning rate set to 6e-5 and the one learning rate cycle scheduler proposed by [159]. In total, we train the segmentation models for 100 epochs with 300 steps per epoch. For all experiments, we use a batch size of 64 during fine-tuning.

Few-Shot Classification on EuroSAT

We additionally illustrate that embeddings learned from GenCo on AV+ help few-shot learning tasks in the more general remote sensing community. To achieve this, we evaluate our proposed method on the few-shot classification task of EuroSAT [57]. Following experiments in previous sections, we evaluate the quality and adaptability of pre-trained features from the proposed methods in this land-cover classification task.

We add one fully connected layer to pre-trained backbones, building the classifier for EuroSAT. We train the model for 100 epochs with an AdamW optimizer and a batch size of 256. The initial learning rate is 0.001.

Table 4.2: Comparison of the classification task between the 10-shot results of GenCo and end-to-end training using the full Agriculture-Vision on ResNet-18.

Pre-trained Weight	Freeze Backbone	Number of Images	Accuracy (%)
Random	False	9,000	57.30 ±0.81
Random	False	94,986	62.31 ±0.25
GenCo on AV+	True	**10**	**65.51**±0.68

Table 4.3: 9-way few-shot classification accuracy on Agriculture-Vision based on weights pre-trained from GenCo.

Backbone \ Shots	Accuracy 10 shots (%)	Accuracy 5 shots (%)	Accuracy 1 shot (%)
ResNet-18	**65.51** ±0.68	**61.61** ±0.71	16.72 ±1.32
ResNet-50	64.82 ±0.70	59.44 ±0.79	**29.64** ±1.28
ResNet-101	64.62 ±0.49	59.56±0.66	28.84 ±1.73

4.3.4 Results of Experiments

Few-shot Learning on Agriculture-Vision

Quality of Pre-Trained Embeddings. We first prove the quality and adaptability of pre-trained embeddings from the proposed methods. As shown in Table 4.1, our pre-trained weights show significantly better results than those from ImageNet, with over 10 points improvement on average. These results prove the adaptability of embeddings encoded from our pre-trained weights and better generalization capacity in this few-shot classification task for agricultural patterns. The best result is obtained from ResNet-18 instead of the larger ResNet-50 or ResNet-101. With only 10 shots, this observation is due to the last layer attached to ResNet-18 being smaller than the fully connected layers in larger backbones. MoCoV2, SimSiam and SimCLR show sub-optimal results compared with GenCo. However, with our two-stage training strategy, all AV+ pre-trained weights enable better performance than any ImageNet weights.

Learning Efficiency. Next, we continue to demonstrate the learning efficiency of GenCo by comparing it with the model trained with 94,986 labeled images. For models training in an end-to-end manner, there

Table 4.4: Comparison of fine-tuning results between weights pre-trained on COCO and weights from GenCo on AV+ for the 10-shot semantic segmentation task.

Pre-trained Weight	Backbone	mIoU - 8 Classes
COCO	ResNet-18	15.61
GenCo on AV+	ResNet-18	**23.56**
COCO	ResNet-50	15.60
GenCo on AV+	ResNet-50	**23.00**
COCO	ResNet-101	15.19
GenCo on AV+	ResNet-101	**21.04**

Table 4.5: Comparison of the segmentation task between the 10-shot results of the proposed method and end-to-end training using the full Agriculture-Vision on ResNet-18 and ResNet-50.

Pre-trained Weight	Backbone	Freeze Backbone	Number of Images	mIoU - 8 Classes
Random	ResNet-18	False	9000	19.02
Random	ResNet-18	False	94986	21.37
SimSiam on AV+	ResNet-18	True	**10**	21.30
SimCLR on AV+	ResNet-18	True	**10**	22.13
MoCo on AV+	ResNet-18	True	**10**	22.11
GenCo on AV+	ResNet-18	True	**10**	**23.56**
Random	ResNet-50	False	9000	19.58
Random	ResNet-50	False	94986	21.82
SimSiam on AV+	ResNet-50	True	**10**	21.06
SimCLR on AV+	ResNet-50	True	**10**	21.98
MoCo on AV+	ResNet-50	True	**10**	21.21
GenCo on AV+	ResNet-50	True	**10**	**23.00**

Table 4.6: Ablation of the proposed generator with different contrastive learning frameworks.

Shots \ Modules	No Generator (MoCo-V2)	Generator with Pre-training	GenCo
10 shot	64.17	64.61	**65.51**
5 shot	59.32	59.91	**61.61**
1 shot	15.21	15.23	**16.72**

is a noticeable drop once we reduce the number of models for training. However, as shown in Table 4.2, GenCo outperforms model training with numerous images with little computation and much fewer labels for agricultural pattern classification. This observation is important as it illustrates the potential of training diverse deep-learning tasks in agriculture and remote sensing with minimum effort but still providing satisfactory results.

Table 4.3 demonstrates the 9-way few-shot classification results with different sizes of backbones. All results are averaged from 3 trials and use the same training setup for a fair comparison. While ResNet-18 gives the best results when trained with five shots or ten shots, ResNet-50 shows the best results when there is only one labeled sample for each class. The performance of ResNet-50 and ResNet-101 are very similar. Generally, favorable results can be acquired when the number of shots is five or greater.

Ablation Study of Few-shot Classification

In this section, we aim to prove the effectiveness of the proposed generator G in the contrastive learning framework. We conduct experiments on GenCo with ResNet-18 as the backbones for pre-training on AV+. We then report the results of the downstream 10-shot classification tasks on the agriculture vision dataset.

We conducted a comparative analysis of three frameworks: MoCo-V2, MoCo-V2 with a generator, and our proposed GenCo. Our experimental results, as summarized in Table 4.6, demonstrate that the use of a generator improves the learned embeddings even when introduced solely during pre-training. This improvement can be attributed to the additional variance and contrast introduced by the positive feature

vectors. However, the performance gain achieved is relatively marginal. Notably, the largest gain occurs when we adopt the pre-trained generator to few-shot learning, which provides additional labeled data in the feature space. As such, the generator can enhance both pre-training and downstream tasks simultaneously, thereby improving the overall performance of the framework.

Few-shot Segmentation on Agriculture-Vision

Quality of Pre-Trained Embeddings. Since U-Net's structures contain skip connections from different layers [153], we don't evaluate a single embedding but features from different scales. Concretely, features from GenCo and features from encoders pre-trained on COCO are compared using the mean intersection over union (mIoU) metric. As reported in Table 4.4, our proposed method shows around 6-8 points of improvement compared with weights pre-trained on COCO. Consistent with the results from the classification task, the best mIoU is reached by ResNet-18 with a smaller decoder attached. The other conclusion we can draw is that the feature distribution pre-trained from natural images (COCO) and remote sensing images (AV+) is significantly different. Therefore, we can observe a noticeable improvement in the results pre-trained on AV+.

Learning Efficiency. We also examine the learning efficiency of the segmentation task. To do this, we use only ten sampled images per category and compare the results of GenCo with those of models trained on the full Agriculture-Vision dataset. While our approach fixes the backbone, we unfreeze the segmentation model's encoder and the generator training on the full dataset. Based on the results presented in Table 4.5, we observe an improvement of 2.19 points and 1.18 points for ResNet-18 and ResNet-50, respectively, using the GenCo approach. However, for MoVo-V2, SimSiam, and SimCLR, the results are comparable or only slightly better than those obtained using the end-to-end training method. Importantly,

while the GenCo-based few-shot segmentation approach still outperforms models trained with a large number of labeled images, we note that the improvement is not as significant as the improvement achieved in the classification task. This observation is likely because the decoders used for segmentation have more parameters to be tuned than a single-layer classifier. With limited labeled samples, smaller models are better able to avoid overfitting and show better results. Therefore, in this few-shot segmentation task, ResNet-18 performs the most satisfactorily.

Few-shot Classification on EuroSAT

Quality of Pre-Trained Embeddings. Results show that GenCo still leads to better embeddings on this remote sensing dataset. As seen in Table 4.7, features from GenCo improve 1% of accuracy on average compared to the features trained from ImageNet. Since EuroSAT shares much less similarity with our pre-trained dataset, i.e., AV+, the improvement is moderate. However, the gain is still stably earned, crossing different sizes of backbones. The results from MoCo-V2, SimSiam and SimCLR still outperform the results of ImageNet but are sub-optimal compared with results from GenCo, proving the effectiveness of the proposed generator.

Learning Efficiency. In the experiments on label efficiency, we continue to compare the few-shot classification models with those models randomly initialized and trained on 27,000 labeled images. The proposed method outperforms the end-to-end model by 3.66 points in this classification task with only ten labeled images, as shown in Table 4.8. This result is crucial as it proves the effectiveness of our methods in different domains. With a remarkably cheap effort of labeling, it re-verifies the vast possibility of deploying our models to various downstream tasks in agriculture and remote sensing.

We show the results of the 10-way few-shot classification on EuroSAT in the following Table 4.9.

Table 4.7: Comparison of fine-tuning results between weights pre-trained on supervised ImageNet and weights from our GenCo on EuroSAT for the 10-shot classification

Pre-trained Weight	Backbone	Accuracy(%)
ImageNet	ResNet-18	66.90 ±0.11
SimSiam on AV+	ResNet-18	67.20 ±0.43
SimCLR on AV+	ResNet-18	67.31 ±0.51
MoCo on AV+	ResNet-18	67.14 ±0.32
GenCo on AV+	ResNet-18	**67.92**±0.31
ImageNet	ResNet-50	65.01 ±0.14
SimSiam on AV+	ResNet-50	65.61 ±0.38
SimCLR on AV+	ResNet-50	65.93 ±0.62
MoCo on AV+	ResNet-50	65.58 ±0.40
GenCo on AV+	ResNet-50	**66.11** ±0.45
ImageNet	ResNet-101	63.34 ±0.20
SimSiam on AV+	ResNet-101	63.32 ±0.35
SimCLR on AV+	ResNet-101	64.17 ±0.68
MoCo on AV+	ResNet-101	63.79 ±0.45
GenCo on AV+	ResNet-101	**64.79** ±0.46

Table 4.8: Comparison of the classification task between the 10-shot results of GenCo and end-to-end training using the full EuroSAT on ResNet-18. *: results referred from [26]

Pre-trained Weight	Freeze Backbone	Number of Images	Accuracy (%)
Random	False	2,700	58.81±0.10
Random	False	27,000	63.21 *
Random	False	27,000	63.34 ±0.08
GenCo on AV+	True	10	**66.90** ±0.31

Table 4.9: 10-way few-shot classification accuracy on EuroSAT based on weights pre-trained from GenCo.

Backbone \ Shots	Accuracy 10 shots (%)	Accuracy 5 shots (%)	Accuracy 1 shot (%)
ResNet-18	**67.92**±0.31	**63.20**±0.37	**11.50**±0.82
ResNet-50	65.01±0.45	59.40±0.51	11.21±1.42
ResNet-101	63.70±0.46	58.70 ±0.66	11.40±1.71

For a complete and fair comparison, we report the performance of backbones with different sizes and average results over experiments with three random seeds. More specifically, the ResNet-18 shows the most satisfactory performance crossing various backbones and shots. While we can notice a 1 to 4 points drop in accuracy when we increase the size of backbones under 10 or 5 shots settings, one-shot classification shows very similar performance regardless of encoder sizes.

Chapter 5

Optimizing Nitrogen Management with Deep Reinforcement Learning and Crop Simulations

5.1 Research Overview

The world urgently needs to move towards more sustainable and resilient cropping systems [160]. Among different factors influencing crop production and the environment, *nitrogen (N) management* is a key controllable one. Nitrogen is the main nutrient affecting crop growth and yield formation, but excessive nitrogen has substantial negative environmental effects [161]. Effective nitrogen management is therefore crucial for maximizing crop yields and farmer income and minimizing negative environmental impacts.

Although best-practice knowledge for N management for common scenarios exists among farmers,

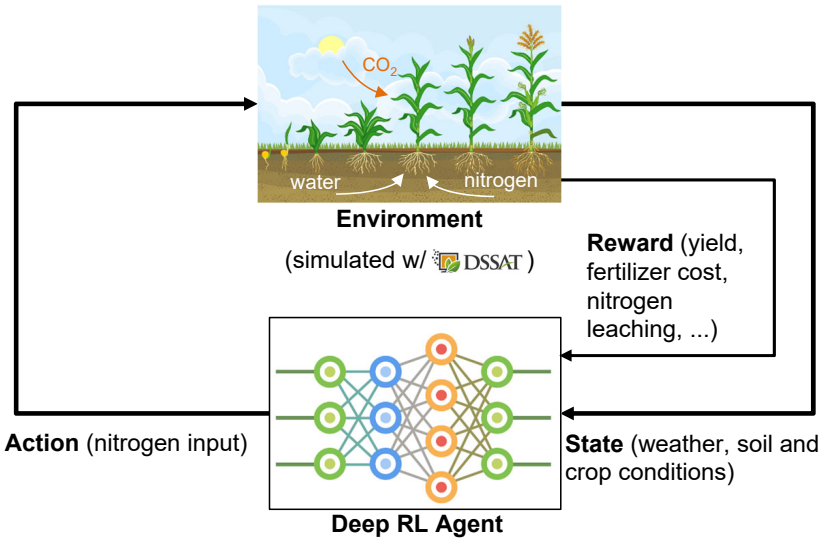

Figure 5.1: A framework for optimizing N management with deep RL and DSSAT-based crop simulations

it is unclear whether these practices are near-optimal, or whether some specific strategies transfer well to adverse seasonal conditions of extreme temperature or precipitation. N management is essentially a sequential decision-making (SDM) problem, as a few decisions on nitrogen application time and quantities need to be made across the growth cycle of crops.

Modern reinforcement learning (RL) methods, represented by deep RL, have achieved remarkable or superhuman performance on a variety of tasks involving SDM such as gaming [126], [127], data center cooling [128], and robotic control [129]–[131]. We expect that RL has a potential for optimizing agricultural management, improving the crop yield while minimizing the environmental impacts. Training a deep RL policy often needs numerous interactions between the RL agent and the environment, which makes it unrealistic to leverage field trial-based approaches [162]. Therefore, training the management policies in simulations [163]–[165], using crop models to simulate the crop and soil dynamics and interact with the RL agent, seems like a realistic solution.

In this work, we *propose and evaluate a framework for optimizing N management using deep RL*

and crop simulations, depicted in Fig. 5.1. In particular, we leverage Decision Support System for Agrotechnology Transfer (DSSAT), a widely used tool for crop modeling and simulation [166], [167], and the Gym-DSSAT interface [139] that allows users to read the simulated crop and soil conditions and apply management practices on a daily basis. As a demonstration of the use of the presented framework, we train N management policies with two deep RL algorithms, namely deep Q-network (DQN) and soft actor-critic (SAC), for the maize crop in Iowa and Florida, US. We further evaluate the performance of the trained policies in comparison with standard practices, and under different scenarios including partial observations and reduced action frequencies.

Compared to early work on RL-based crop management [124], [125], our framework, which leverages deep RL, can handle much larger state and action spaces. Compared to recent work on deep RL-based agricultural management [132], [133], the crop model adopted in our framework, i.e., DSSAT, is much more widely used globally; additionally, our experimental study is significantly more comprehensive, which involves two different deep RL algorithms, two geographic locations, and ablation study for partial observations and reduced action frequencies.

5.1.1 MDP Problem Formulation

The N management problem can be formulated as a finite Markov decision process (MDP) problem. In this formulation, a decision-making agent continuously interacts with the environment. At each day t, the agent selects an action (i.e., management practice), a_t, from the action space $\mathcal{A}$, based on the current state s_t, which is an array of elements from the state space $\mathcal{S}$. The selected action is applied to the environment and a new state (s_{t+1}) is generated based on this action; meanwhile, a reward signal $r_t = r(s_t, a_t)$ is produced to evaluate the immediate consequence of the selected action. This interaction repeats until the

termination of the interaction, e.g., when the crop is harvested. The goal of the agent is to select optimal actions to maximize the future discounted return. The future discounted return at time t is defined as $R_t = \sum_{\tau=t}^{T} \gamma^{\tau-t} r_\tau$, where T is the time step at termination.

For N management, the action space $\mathcal{A}$ contains all possible amounts of nitrogen applied at a day. All the states that compose the state space are listed in Table 5.7. The reward function $r(s_t, a_t)$ at day t is set as:

$$r(s_t, a_t) = \begin{cases} w_1 Y - w_2 a_t - w_3 N_{l,t} - w_4 P_t & \text{if harvest at } t, \\ -w_2 a_t - w_3 N_{l,t} - w_4 P_t & \text{otherwise,} \end{cases} \tag{5.1}$$

where a_t is the action (i.e. amount of nitrogen applied at day t), $N_{l,t}$ is the nitrate leaching at day t, Y is the crop yield at the harvest date represented by the top weight at maturity, and P_t is the additional penalty on large *total amount* of nitrogen applied. In particular, $P_t = \sum_{k=1}^{t} a_k - \mathit{threshold}$ if $a_t \neq 0$ and $P_t = 0$ if $a_t = 0$, where *threshold* represents the allowable total amount of nitrogen inputs. It may be worth mentioning that nitrate leaching occurs when nitrate is washed out of the root zone by heavy rainfall. Leaching is undesirable because it leads to the waste of the fertilizers, and more importantly, causes environmental problems such as eutrophication of watercourses and soil degradation. Thus, we include a penalty on nitrate leaching in the reward function. Finally, the positive constants $w_1 \sim w_4$ are selected to balance the different aspects mentioned above.

5.2 Training Management Policies using Deep RL

For solving the formulated MDP problem, we leverage the recently proposed deep RL algorithms, which have achieved remarkable performance on a variety of tasks [126]–[131], [168]. We choose deep Q-network (DQN) [126] and soft actor-critic (SAC) [169] for the experimental study, but other deep RL algorithms

capable of handling continuous state spaces can also be applied.

5.2.1 Policy Training with DQN

DQN is a model-free deep RL algorithm, which uses a deep neural network (DNN) to approximate the action-value function in Q-network [126]. The essential idea of DQN is to learn an optimal action-value function $Q^\star(s,a) = \max_\pi \mathbb{E}[R_t|s_t = s, a_t = a, \pi]$, where π is a policy mapping a state s_t to an action a_t at a given time t. With the $Q^\star$ function, given an action s_t, an optimal action $a_t^\star$ can be determined, e.g., by following a greedy policy defined by $a_t^\star = \max_{a\in\mathcal{A}} Q^\star(s_t, a)$. From the interaction between the agent and environment, tuples of (s, a, r, s') are generated and stored in a replay buffer, where s, a, r and s' denote current state, current action, immediate reward obtained by applying the action a at the state s, and next state, respectively. Due to the nature of continuity of the action space, we discretized the action space. At iteration i, the Q network can be trained by minimizing the loss function:

$$L_i(\theta_i) \triangleq \mathop{\mathbb{E}}_{(s,a,r,s')}\left[r+\gamma \max_{a'\in\mathcal{A}} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i)\right], \tag{5.2}$$

where the tuples (s, a, r, s') are sampled from the replay buffer, θ_i are the parameters of the Q-network at iteration i, and θ_i^- are the network parameters used to compute the target at iteration i. The optimization problem can be solved using stochastic gradient descent algorithms [126].

5.2.2 Policy Training with SAC

SAC is a policy-gradient deep RL algorithm that represents the state of the art among model-free RL algorithms in terms of sample efficiency and stability with respect to the hyperparameters [169]. Besides the expected sum of rewards, SAC introduces the expected entropy to favor stochastic policies, which leads

to a cost function L defined by

$$L \triangleq -\sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim p_\pi} \left[r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \right], \tag{5.3}$$

where p denotes the state-action marginals of the trajectory distribution, $\mathcal{H}$ determines the entropy for the evaluation of randomness given the state s_t, and $r(s_t, a_t)$ is the immediate reward at time t. The temperature parameter α decides the trade-off between the entropy term and rewards.

5.3 Simulating the Crop Response using Gym-DSSAT

DSSAT has been used for various crop simulations worldwide in the last 30 years [166]. However, limited interactions can be reached during the running period of simulation, leading to a possible delay of adjustment for management decisions. Recently, Gym-DSSAT [139] has been developed to bridge the communication gap between the simulation environment and daily management decisions. This communication pipeline enables RL researchers to manipulate DSSAT like Open AI Gym in machine learning and robotics [126], [127]. In Gym-DSSAT, the environment is defined at a field scale with a time step corresponding to one day. An episode typically covers about 160 days from planting to harvest, and its state is automatically set as "done" at crop maturity. Weather is randomly generated via WGEN's [170] built-in stochastic weather generator and can be fixed depending on simulation purposes.

With Gym-DSSAT, millions of daily interactions between an RL agent and the simulated crop environment can be achieved in a few minutes, and used for training the management policies.

5.4 Experiments and Results

We conducted experiments on training N management policies for the maize crop in both Florida and Iowa. These two locations are selected since they have different weather and soil conditions, which can be leveraged to test the general applicability of the proposed framework. Also, DSSAT includes templates for simulating the maize crop in these two locations, which facilitates the implementation of our proposed framework. We evaluated the performance of trained policies in comparison with the standard practice proposed in [171].

5.4.1 Datasets for Florida and Iowa

Two experiments were studied. The first one is for the maize crop in Ames, Iowa, in 1999. The simulation starts on April 25th, the planting happens on May 27th, and the crop is harvested no later than Oct 24th. The soil has a depth of 151 cm, and the plant density is 7.6 plant/m^2. The second experiment is for the maize crop in Gainesville, Florida, in 1982. In the Florida setup, the simulation starts on Jan 30th, while the crop is planted on Feb 26th and harvested when reaching maturity. The soil in this case has a depth of 180 cm, and the plant density is 7.2 plant/m^2. For both simulations, the irrigation is set to 0. This is consistent with the current practice in Iowa, where the maize crop is not irrigated. On the other hand, irrigation is crucial to improve the maize yield in Florida in reality. Setting the irrigation to 0 for the Florida case can be considered as an emulation of the extreme case of severe drought and limited water supply, which allows us to compare the RL-based management strategy with the standard practice under this extreme case.

5.4.2 Implementation Details

For all the experiments, weight parameters w_1, w_2, and w_3 in the reward function (5.1) were set to be 0.1, and w_4 is set to be 1. For both DQN and SAC, we implemented the training using Pytorch, and used the Adam [172] optimizer with an initial learning rate of 0.00005 and a batch size of 64 to train the neural network. We trained the policies for 1200 episodes with the exploration rate ϵ decreasing from 1 to 0, following a decay factor of 0.994 for the Florida case and of 0.992 for the Iowa case.

For DQN, the discrete action space was defined to be $\mathcal{A} = \{40k\frac{\text{kg}}{\text{ha}} | k = 0, 1, 2, 3, 4\}$. The discount factor was set to be 0.99.

For SAC, the agent action a_{sac} varies from 0 to 200 and is discretized into the same action space as the one used for DQN through the mapping: $\arg\min_{a \in \mathcal{A}} \quad \|a_{sac} - a\|$, for both training and testing. The discretization is for being consistent with farmers' fertilization patterns, i.e., fertilize only a few times in the whole growth cycle. The discount factor and smoothing constant for updating the target network were set to be 0.98 and 0.001, respectively.

For comparison with the trained policies, we also implemented the standard management practice in [171], which suggests to add nitrogen at vegetative growth stage (vstage) 5, the stage when crop reaches five expanded leaves.

5.4.3 Results for the Iowa Maize

The training curve using DQN, averaged over five trials, is shown in Fig. 5.2. During the first 200 episodes of exploration, due to the large exploration rate, the DQN agent over-fertilizes causing significant penalties. After 800 episodes of training, the learning converges, constantly giving a cumulative reward of over 2000. Concretely, Table 5.1 compares the performance of the DQN-trained policy and three baseline

strategies, corresponding to 160, 240, 280 kg/ha of nitrogen applied at stage v5, as suggested in [171]. The trained DQN agent decides to apply a total of 240 kg/ha nitrogen input during the growing season, and achieves 21711.8 kg/ha top weight of maize at maturity and a cumulative reward of 2126.3. Among three baseline policies, the one with 280 kg/ha achieves the largest cumulative reward of 2142.9 and largest top weight of 21709.5 kg/ha. Compared with the best baseline, DQN achieves slight improvement on top weight at maturity while using 14% less nitrogen input, being more cost-efficient. Compared to the baseline using same amount of nitrogen input, DQN achieves a 1% increment on the top weight at harvest. In general, the trained DQN agent achieves better results than the baseline methods.

The performance of SAC is shown in Table 5.2. Although using less nitrogen that causes a smaller top weight at maturity, the SAC policy still achieves a cumulative reward similar to that achieved by the DQN policy. Thus, both RL algorithms succeeded in finding a better management policy than the baselines. However, the learning process converged much faster with SAC. Specifically, the cumulative reward with SAC reached 2100 within 700 episodes, while additional 300 episodes were needed to achieve similar results with DQN.

Table 5.1: Performance comparison between DQN and baseline policies for Iowa. Baseline (X) indicates that X kg/ha of nitrogen is applied at stage v5.

Methods	Nitrogen input (kg/ha)	Nitrate leaching (kg/ha)	Nitrogen uptake (kg/ha)	Top weight at maturity (kg/ha)	Cumulative reward
Baseline (160)	160	0.11	219.1	21133.3	2097.3
Baseline (240)	240	0.11	264.0	21502.9	2126.3
Baseline (280)	280	0.11	272.3	21709.5	2142.9
DQN	240	0.12	290.4	**21711.8**	**2147.1**

Table 5.2: Performance comparison between SAC and DQN policies for Iowa.

Methods	Nitrogen input (kg/ha)	Top weight at maturity (kg/ha)	Cumulative reward	Episodes of convergence
DQN	240	21711.8	2147.1	1000
SAC	200	21503.3	2144.2	700

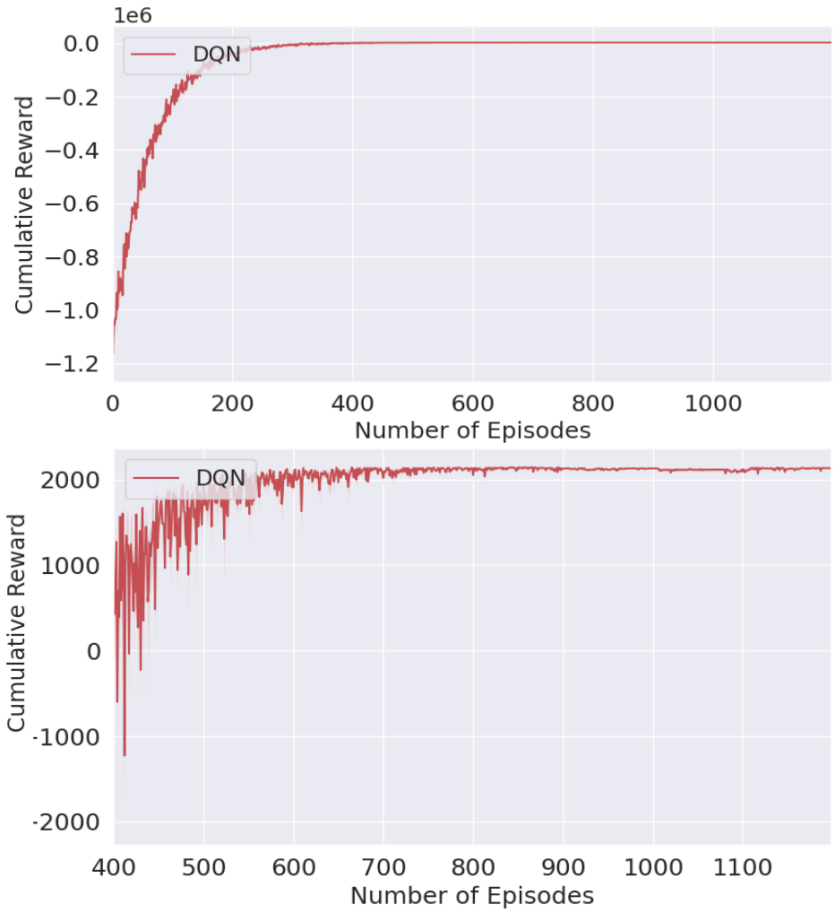

Figure 5.2: Cumulative reward versus episodes with DQN for Iowa. Results are averaged over five trials, with the light-red shaded area denoting the variance. Top: full view. Bottom: zoomed-in view for 400–1200 episodes

5.4.4 Results for Florida Case

As we mentioned in 5.4.1, the Florida case is not realistic due to the 0 irrigation setup, and can be considered as an extreme weather case under severe drought with water shortage. Accordingly, the yields obtained under this setup are much smaller compared to those in the Iowa case. The training curve under DQN averaged over five trials is shown in Fig. 5.3. The performance comparison between our trained DQN policies and baselines is summarized in Table 5.3. As one can see, the DQN policy shows a stable improvement in terms of the top weight and rewards, which is consistent with the results for Iowa. The performance comparison between SAC and DQN is shown in Table 5.4. Similar to the Iowa case, the SAC

policy achieved a similar cumulative reward as the DQN policy but the training with SAC converged much faster.

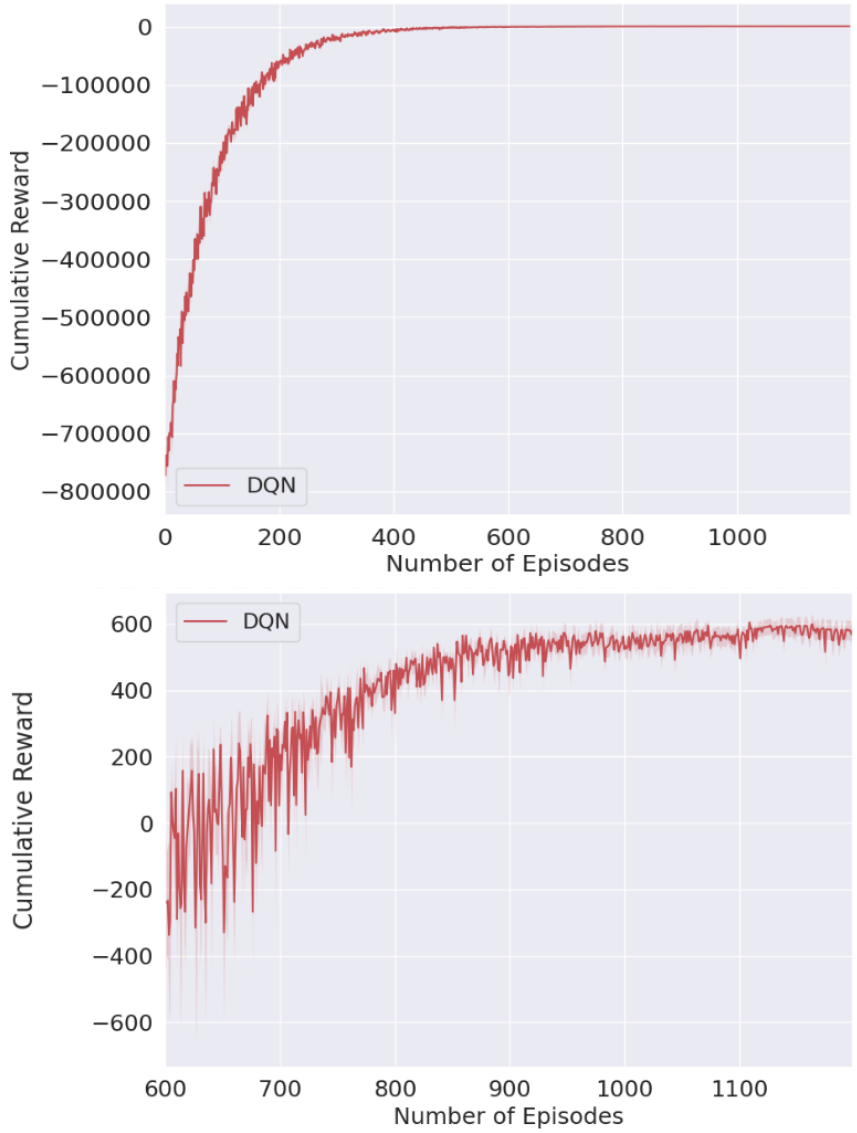

Figure 5.3: Cumulative reward versus episodes with DQN for Florida. Results are averaged over five trials, with the light-red shaded area denoting the variance. Top: full view. Bottom: zoomed-in view for 400–1200 episodes

Table 5.3: Performance comparison between DQN and baseline policies for Florida. Baseline (X) indicates that X kg/ha of nitrogen is applied at stage v5.

Methods	Nitrogen input (kg/ha)	Nitrate leaching (kg/ha)	Nitrogen uptake (kg/ha)	Top weight at maturity (kg/ha)	Cumulative reward
Baseline (40)	40	46	55	4393.3	430.7
Baseline (80)	80	65	66	4673.1	452.8
Baseline (160)	160	97	86	5190.4	493.3
DQN	80	33	105	**6310.8**	**619.7**

Table 5.4: Performance comparison between SAC and DQN policies for Florida.

Methods	Nitrogen input (kg/ha)	Top weight at maturity (kg/ha)	Cumulative reward	Episodes of convergence
DQN	80	6310.8	619.7	900
SAC	80	6308.0	610.2	700

5.4.5 Ablation Study

In practice, not all the states used in the training and testing of the management policies in the previous sections are accessible. Additionally, from an economic perspective, farmers prefer to make decisions less frequently, e.g., weekly and biweekly, instead of daily. Therefore, in this section, we study the effect of full/partial observation and action frequencies on the performance of the proposed framework.

Full vs. Partial Observation

To understand the contribution of observed states in the fertilizer optimization process, we carry out an ablation study on (i) full observation case, in which all the states listed in Table 5.7 are used for policy training and testing and (ii) partial observation case, in which only 10 states (indicated in Table 5.7) are used. The study on partial observation is motivated by the fact that not all the states output by DSSAT can be accessed by farmers without professional agricultural tools for detection and inspection. Experiments of full observation have been conducted in Section 5.4. The results under partial observation, which are based on DQN, are shown in Fig. 5.4. For both Florida and Iowa, the policy training and testing under partial observation were conducted three times, and Fig. 5.4 shows the results averaged over the three trials.

For both Florida and Iowa setups, the policies under partial observation always underperformed those under full observation. In particular, we observed 30.15% and 3.58% decreases in reward and 27.94% and 3.68 % drops in the final yield for Florida and Iowa, respectively. The decrease for Florida is relatively large compared to that for Iowa. This could be attributed to the extreme weather condition associated with

Florida which requires the RL agent to have more comprehensive information to make a good decision.

Action Frequency

We ablate the action frequency in the life cycle of maize to further understand the applied actions during the simulation process. We continue to conduct all the experiments with DQN in a discrete space to ensure the consistency with farmers' fertilization patterns. Concretely, we experiment with the trained DQN policy using two action frequencies: (i) RL agents are allowed to fertilize every day and (ii) RL agents are only permitted to fertilize every ten days.

For Iowa, Fig. 5.5 shows the applied actions and Table 5.5 lists achieved cumulative reward and top weight at maturity under different actions. The DQN agents fertilized 5 to 6 times. Both the baseline method and the DQN policy with a 10-day action frequency used 280 kg/ha nitrogen input. However, their results are relatively poor compared with the DQN policy with a 1-day action frequency.

The results for Florida are shown in Fig. 5.6 and Table 5.6. One can see that both baseline methods and our RL agent chose to fertilize once across the crop's life cycle. However, the RL agent tended to apply less nitrogen, as shown in Fig. 5.6. According to Table 5.6, the performance of the DQN agent degrades a little bit under the reduced action frequency of 10 days, but is still better than that of the best baseline.

Table 5.5: Performance of trained policies under different action frequencies for Iowa. DQN (1 Day): RL agents are allowed to fertilize every day. DQN (10 Days): RL agents are only permitted to fertilize every ten days.

Method	Top weight at maturity (kg/ha)	Cumulative reward
DQN (1 Day)	21710.7	2147.1
DQN (10 Days)	21700.6	2142.0
Baseline	21709.5	2142.9

Table 5.6: Performance of trained policies under different action frequencies for Florida.

Method	Top weight at maturity (kg/ha)	Cumulative reward
DQN (1 Day)	6310.8	619.8
DQN (10 Days)	5728.9	565.7
Baseline	5190.4	493.3

Table 5.7: State space description

State	Description	Included in Partial Observation Study?
cumsumfert	cumulative nitrogen fertilizer applications (kg/ha)	✓
dap	days after simulation started	✓
dtt	growing degree days for current day (C/d)	✓
istage	DSSAT maize growing stage	✓
vstage	vegetative growth stage (number of leaves)	✓
pltpop	plant population density (plant/m2)	✓
rain	rainfalls for the current day (mm/d)	✓
srad	solar radiations during the current day (MJ/m2/d)	✓
tmax	maximum temparature for current day (C)	✓
tmin	minimum temparature for current day (C)	✓
sw	volumetric soil water content in soil layers (cm3 [water] / cm3 [soil])	✓
nstres	index of plant nitrogen stress (unitless)	
pcngrn	massic fraction of nitrogen in grains (unitless)	
swfac	index of plant water stress (unitless)	
tleachd	daily nitrate leaching (kg/ha)	
grnwt	grain weight dry matter (kg/ha)	
cleach	cumulative nitrate leaching (kg/ha)	
cnox	cumulative nitrogen denitrification (kg/ha)	
tnoxd	daily nitrogen denitrification (kg/ha)	
trnu	daily nitrogen plant population uptake (kg/ha)	
wtnup	cumulative plant population nitrogen uptake (kg/ha)	
xlai	plant population leaf area index (m2_leaf/m2_soil)	
topwt	top weight (kg/ha)	
es	actual soil evaporation rate (mm/d)	
runoff	calculated runoff (mm/d)	
wtdep	depth to water table (cm)	
rtdep	root depth (cm)	
totaml	cumulative ammonia volatilization (kgN/ha)	

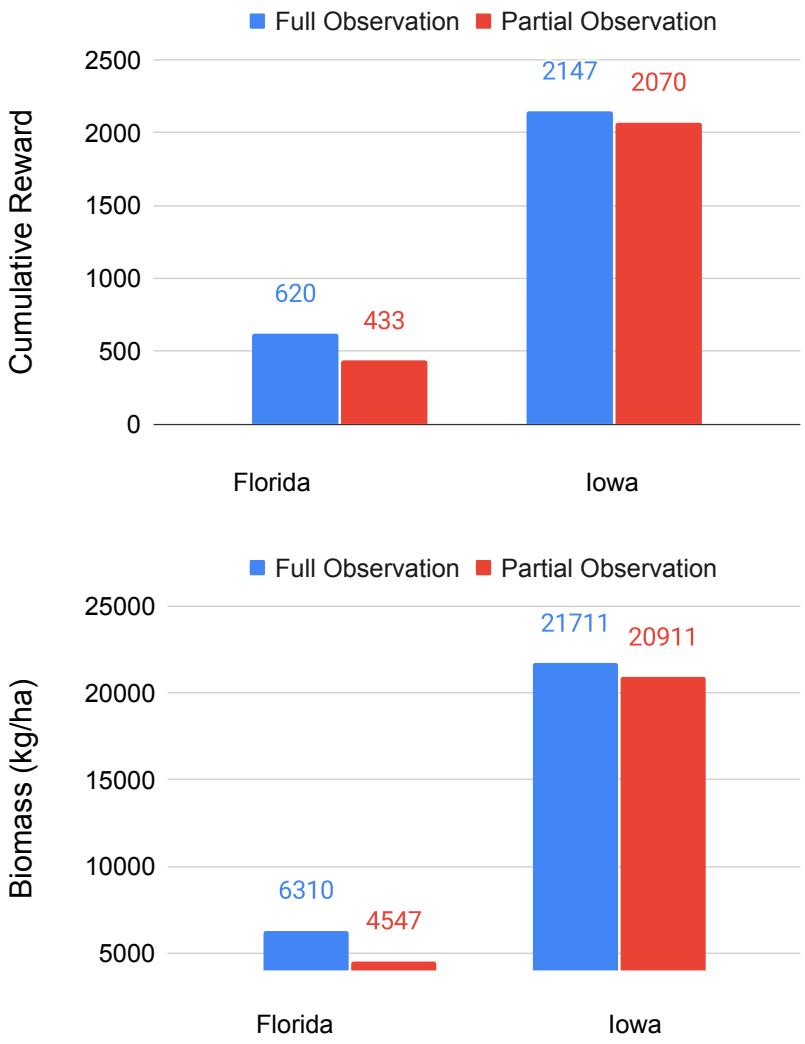

Figure 5.4: Comparison of cumulative rewards (top) and final above the ground population biomass (bottom) obtained under partial observation and full observation. The results are averaged over three trials.

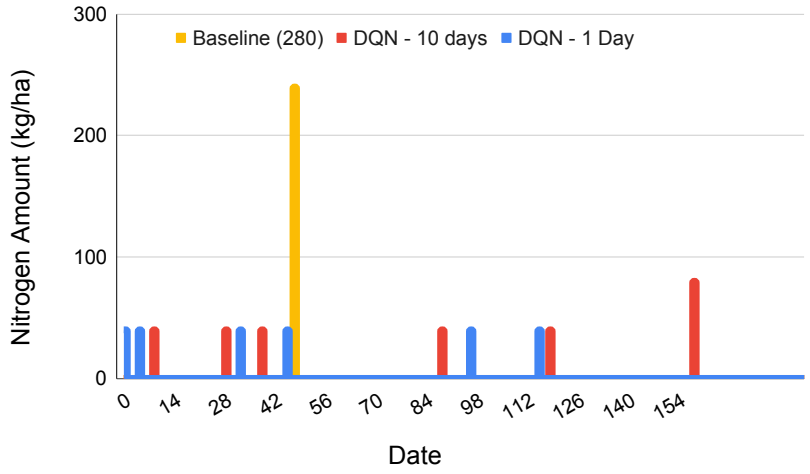

Figure 5.5: Applied actions under different action frequencies for Iowa.

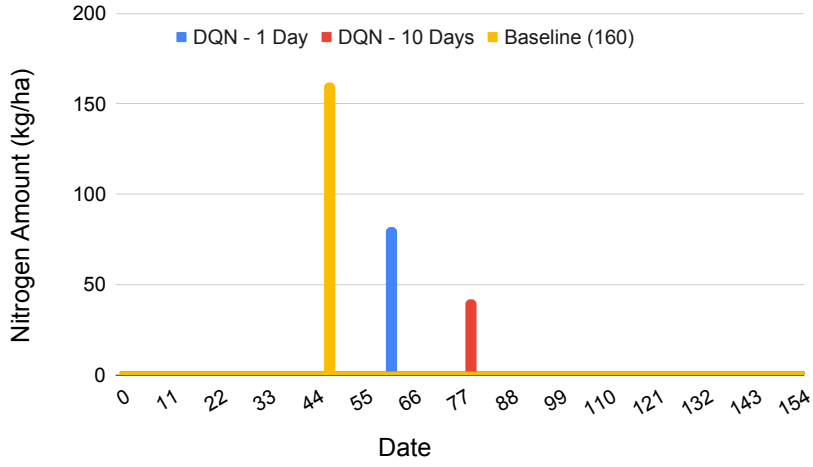

Figure 5.6: Applied actions under different action frequencies for Florida.

Chapter 6

The New Agronomists: Language Models are Experts in Crop Management

6.1 Research Overview

Key factors in crop management, particularly fertilization with nitrogen (N) and irrigation with water (W), significantly affect crop yields and environmental health [173]. However, the previous best practices for these management aspects, derived from empirical experience and academic research [174], [175], face uncertainty in their effectiveness against changing climate and market conditions. Therefore, the adequacy of current strategies is questionable, highlighting a need for innovative, efficient, and adaptable management systems. These systems should be capable of devising optimal strategies suitable for varying conditions and

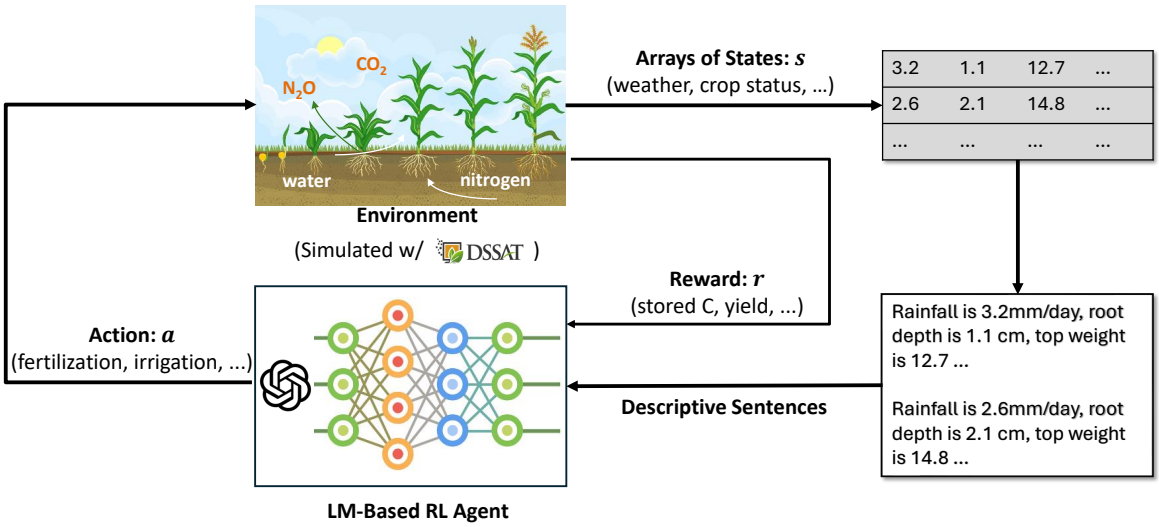

Figure 6.1: Framework and pipeline of the intelligent crop management system using LM-based RL

objectives, such as maximizing economic profit [176]. This research is anchored in this context, leveraging advanced AI methods to improve agricultural practices and tackle these critical challenges.

Reinforcement learning (RL) has shown exceptional capabilities in tasks that involve sequential decision-making (SDM), such as in robotics and gaming [126], [129]. This success suggests a significant potential for RL in optimizing crop management, which at its core is an SDM problem. Given the need for numerous interactions between the RL agent and the environment during policy training, field trial-based methods are impractical. Consequently, the use of crop models to simulate both the crop and its environment, providing a platform for interaction with the RL agent, appears to be the most feasible approach [163], [164].

Recently, the authors of [85], [86] proposed to train management policies for crop management using deep RL with DSSAT [166] and Gym-DSSAT [177], one of the most widely used crop models in the world. Their trained policies, both under full and partial observations, outperformed baseline policies by achieving higher yields or similar yields with reduced nitrogen (N) fertilizer input. However, there are limitations to these approaches. Firstly, the models primarily employed Multilayer Perceptrons (MLPs), which,

while effective, have limited fitting power compared to more complex architectures. This limitation could potentially constrain the models' ability to capture the intricate dynamics of crop growth and management fully. Secondly, the reliance on MLPs limits the incorporation of additional descriptive features for state representations in the model. These features could include various environmental, soil, and crop growth parameters that are crucial for precise agricultural decision-making. This gap in the model's design raises a critical question: Can language models (LMs) serve as viable alternatives for RL agents in these crop management tasks?

We present an intelligent crop management framework, depicted in Figure 6.1, that incorporates a powerful LM, and crop simulations via DSSAT and Gym-DSSAT. Concretely, we transform the states from simulation tools, typically arrays of numbers, into more descriptive sentences. This conversion enables a significant shift in our approach: we replace the traditional MLP-based RL agent with an LM-based RL agent. This new agent leverages LMs to encode these descriptive state sentences into embeddings, thereby capturing a more informative and nuanced understanding of the states. Meanwhile, we notice that LMs have shown distinctive cognitive capabilities, which include advanced thinking [178], robust memory functions [179], and reflective skills [180]. As a result, the RL agent should be equipped with the ability to comprehend complex aspects of crop growth and simulation environments. We, therefore, anticipate that the incorporation of LMs will markedly improve the performance of the RL agent in crop management tasks.

To demonstrate the effectiveness of the proposed method, we conduct case studies simulating maize crop management in Florida, USA, and Zaragoza, Spain. This choice of locations aligns with the settings used in previous studies [86]. In both scenarios, the policies trained by our framework exhibited superior performance compared to the previous state-of-the-art approaches; the baseline was derived from either

maize production guidelines recommended by agricultural experts as well as survey results on actual management practices of maize farmers. Additionally, continuing in the vein of established research, our investigation also includes the training of RL-based policies with well-recognized reward functions [86]. These functions are designed to represent different balances among key factors: crop yield, resource utilization, and environmental impact, particularly focusing on nitrate leaching during the crop growth cycle. In summary, the primary contributions of our work can be delineated as follows:

- We investigate a critical yet under-explored question: Can LMs serve as better alternatives for RL agents in crop management tasks to offer more nuanced and effective solutions and advance the state of intelligent crop management systems?
- To the best of our knowledge, this work marks the first attempt to integrate descriptive language to represent agricultural states and to employ LMs in the pursuit of optimal crop management policies.
- We empirically demonstrate that our proposed framework exceeds the performance of existing state-of-the-art approaches in various key aspects, including crop yield, resource utilization, and environmental impact.

6.2 Methods

6.2.1 Problem Formulation

In this study, we approach nitrogen fertilization and irrigation management as a finite MDP, following the paradigm of previous work [85], [86]. Each day, denoted as day t, involves the agent receiving the environmental state, s_t, and subsequently selecting an action a_t from the action space $\mathcal{A}$. This selection is guided by a policy $\pi(s_t, \theta_t)$, where θ_t symbolizes the policy parameters on that particular day, and notably,

the policy in this context is a pretrained language model. The state s_t encompasses vital data pertaining to weather, plant growth, and soil conditions, as simulated for that day. The action a_t is composed of two key decisions: the quantity of nitrogen fertilizer, denoted as N_t, and the amount of irrigation water, W_t, to be applied. The effectiveness of these decisions is quantified by the reward $r_t(s_t, a_t)$, which is calculated based on the outcomes of s_t and a_t, defined as:

$$r_t(s_t, a_t) = \begin{cases} w_1 Y - w_2 N_t - w_3 W_t - w_4 N_{l,t} & \text{if harvest at } t, \\ -w_2 N_t - w_3 W_t - w_4 N_{l,t} & \text{otherwise,} \end{cases} \tag{6.1}$$

where $w_1, w_2, w_3, w_4, Y, N_{l,t}$ denote four custom weight factors, yield at harvest and the amount of nitrate leaching on a given day, respectively. Both Y and $N_{l,t}$ are derived from the state variable s_t. The design of the reward function, characterized by the weights w_1, w_2, w_3, w_4, is pivotal in steering the agent's strategy. The agent's objective is to identify the optimal policy $\pi(s_t, \theta_t)$ that selects action a_t to maximize the total future discounted return. This return, defined as $R_t = \sum_{\tau=t}^{T} \gamma^{\tau-t} r_\tau$, captures the accumulated reward from the current action a_t to the future rewards, discounted by factor γ.

6.2.2 LM-based RL Agent

To harness the full potential of LM in comprehending crop models and identifying optimal management strategies, we made adaptations to the state variables from the simulation tool, specifically Gym-DSSAT. Traditionally, the state in such simulations is represented by an array of variables reflecting various crop and environmental conditions, like rainfall and root depth. However, this format does not provide a direct correlation between the variables and their descriptive meanings, posing a challenge for RL agents to interpret each variable independently. To overcome this, we transformed the raw data into a more

language-friendly format. Each variable name and its corresponding value were combined into coherent sentences. This approach essentially transforms the state data into a format that is more accessible and interpretable by LMs, allowing for a more intuitive and efficient exploration of management practices.

In our approach, we have innovated by substituting the traditional MLPs with a distilled and pre-trained BERT model from [181] serving as the RL agent. This advanced model is utilized to encode the concatenated sentences, which represent the state variables, into feature embeddings. Following this encoding process, we introduce a few fully connected layers connected to the distilled BERT encoder. These layers are responsible for transforming the generated feature embeddings into a format that aligns with the action space of the RL agent. This novel architecture not only leverages the linguistic understanding of BERT but also ensures that the complex relationships within the crop management data are effectively captured and translated into actionable insights.

6.2.3 Policy Training with LM

In this study, we use the Deep Q-Network (DQN) from [126] to train our agent. The goal is to learn an optimal policy that maximizes the future discounted return, denoted as R_t. A novel aspect of our approach is the integration of the distilled BERT model to represent the action-value function, also known as the Q function, within the DQN framework. This Q function, formally defined as $Q^{\pi}(s, a) = \mathbb{E}[R_t | s_t = s, a_t = a, \pi]$, is essential for calculating the expected future discounted return from state s when action a is taken, following policy π.

The objective is to refine the parameters of the Q-network to pinpoint the optimal Q function, $Q^{\star}(s, a)$, which indicates the highest return possible from state s by taking action a and adhering to the optimal policy. For selecting the optimal action in state s_t, we employ a greedy policy defined as $a_t^{\star} = \max_{a \in \mathcal{A}} Q^{\star}(s_t, a)$.

Training the Q-network, which effectively means training the policy, involves minimizing the following loss function:

$$L_i(\theta_i) \triangleq \mathop{\mathbb{E}}_{(s,a,r,s')} \left[r + \gamma \max_{a' \in \mathcal{A}} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right]. \tag{6.2}$$

Here, s, a, r, s' denote the state, action, reward, and next state, respectively, with γ representing the discount factor, and θ_i^- representing the parameters of a target network defined earlier. The tuples (s, a, r, s') for the loss function are randomly sampled from the replay buffer, a collection of prior state-action-reward-next state tuples accumulated during training.

6.2.4 Crop Simulations with Gym-DSSAT

Similar to [85], [86], we leverage Gym-DSSAT [177], a Gym interface for DSSAT that enables the agent to interact with the simulated environment (i.e., reading the weather, soil, and crop information and applying management practices) on a daily basis.

6.3 Experiments and Results

In this section, various experiments are conducted on real-world datasets to demonstrate the effectiveness and superiority of the proposed framework. The experimental settings are introduced in Section 6.3.1. Following this, the training and evaluation details are illustrated in Section 6.3.2, providing the necessary details to reproduce the results. Then, we present the evaluation results, where the performance of our proposed method is compared against existing baselines and SoTA approaches in Section 6.3.3. Lastly, ablation studies are conducted for policy training in Section 6.3.4.

6.3.1 Experimental Setup

The experiments focusing on training policies for nitrogen and irrigation management in maize crops were conducted through two distinct case studies, both utilizing real-world data. The first of these case studies was set in a simulated environment replicating Florida, USA, in 1982, while the second case study was based on the simulated conditions of Zaragoza, Spain, in 1995. The primary objective of these case studies was to test and demonstrate the viability and advantages of the proposed framework, rather than preparing it for immediate real-world application. For those interested in the specifics of deploying this framework in practical settings, further details are provided in Section 6.4.

For each case study, DQN was used to train the LM-based RL agent under full observation. The performance of all trained policies was evaluated in simulation, and compared with baseline policies and previous state-of-the-art methods as mentioned in [86]. The baseline for the Florida study was based on a maize production guide for Florida farmers [174], and for the Zaragoza study it was derived from survey data on maize farming practices in Zaragoza [175], [182].

The framework was implemented to train the RL agent under full observation. This approach involved testing with five different reward functions, each designed to demonstrate the adaptability of the framework to various trade-offs. These trade-offs include balancing crop yield, N fertilizer use, irrigation water use, and environmental impact. This variety in reward functions allows the framework to be evaluated across a range of scenarios and objectives, showcasing its flexibility in addressing different agricultural management priorities.

6.3.2 Implementation Details and Evaluation Metrics

Implementation Details. The RL agent in our study employs a combination of DistilBERT and a three-layer fully connected neural network for feature adaptation. The process begins with DistilBERT encoding the state inputs into 768-dimensional embeddings. Notably, the parameters of DistilBERT are trained end-to-end in this model. After this initial encoding, the embeddings are passed through fully connected layers, one with 512 units and the other with 256 units. The final layer in this sequence is responsible for mapping these processed embeddings to the action space, completing the flow from the input state to the actionable output in the RL framework. The discrete action space is defined as follows:

$$\mathcal{A} = \{40k\frac{\text{kg}}{\text{ha}} \text{ N fertilizer } \ \& \ 6k\frac{\text{L}}{m^2} \text{ Irrigation water}\}, \tag{6.3}$$

where $k = 0, 1, 2, 3, 4$, resulting in a total of 25 possible actions. This action space design incorporates standard quantities of N fertilizer and irrigation water that are typically applied by farmers in a single day. It also allows for a wide range of options, aiding the discovery of effective policies. The discount factor is meticulously set at 0.99. To facilitate the neural network's updates, Pytorch is employed alongside the Adam optimizer [172], characterized by an initial learning rate of 1e-5 and a batch size of 512. This setup is strategically chosen to optimize the learning process while ensuring efficient computation.

The direct application of DistilBERT's tokenizer to numerical values introduces significant training instability. Concretely, numerical values are often segmented into multiple tokens, resulting in considerable variance for small numerical differences. For instance, the number 360 tokenizes into [9475], while 361 splits into [4029, 2487], indicating a disproportionate representation of adjacent numbers. This inconsistency can

	w_1 (Y)	w_2 (N_t)	w_3 (W_t)	w_4 ($N_{l,t}$)	Note
RF 1	0.158	0.79	1.1	0	Economic profit
RF 2	0.158	0.79	0	0	Free water
RF 3	0.158	0	1.1	0	Free N fertilizer
RF 4	0.158	1.58	1.1	0	Doubled N price
RF 5	0.2	1	1	5	With N Leaching

Table 6.1: Weights used in each reward function (RF) defined by equation (5.1)

amplify instability during training. Additionally, the tokenization of decimal numbers compounds this issue. For example, 0.1 translates into [1014, 1012, 1015], where '0' and the decimal point are tokenized separately, leading to unnecessary token proliferation and computational inefficiency.

To address the tokenization challenges with numerical values in our model, we have developed a straightforward yet effective data preprocessing technique. This method involves normalizing numerical values to fit within the range of [0, 300] and subsequently utilizing only the integer portion for tokenization. The decision to cap the range ensures that each normalized number corresponds to a single token, thereby simplifying and stabilizing the tokenization process. Additionally, focusing solely on the integer part helps to minimize the number of tokens used. We achieve a succinct representation comprising 27 distinct tokens, which includes 25 feature-specific tokens plus two special tokens ([CLS] and [SEP]). This streamlined token set not only improves the stability of the training process but also enhances its computational efficiency, which is crucial for the complex task of crop management optimization using RL and language models.

Evaluation Metrics. In each case study, we employed reward functions in line with the approach described in previous research [86]. Specifically, five distinct reward functions for r_t derived from Equation (5.1) were utilized to train the RL agent. For each reward function, a single trained policy was selected for evaluation. The parameters for each reward function (RF) are detailed in Table 6.1.

RF1 quantifies the economic profit ($/ha) that farmers accrue, calculated based on the estimated prices of maize and the costs of N fertilizer and irrigation water, as referenced from [183] and [174].

RF2-RF4 represent variations of economic profit under different scenarios. Specifically, RF2 addresses the hypothetical situation where irrigation water is free of cost; RF3 considers the case where N fertilizer is free; and RF4 models a scenario in which the price of N fertilizer is doubled.

In contrast to RF1-RF4, which focus solely on economic profit, RF5 incorporates an additional environmental aspect, specifically nitrate leaching. Nitrate leaching is a significant environmental concern as it contributes to problems like eutrophication of water bodies and soil degradation [184]. RF5 is structured to balance yield, N fertilizer, and irrigation use while assigning a substantially higher weight to nitrate leaching. This approach aims to minimize nitrate leaching while still achieving favorable economic outcomes.

Table 6.2: The evaluation results of our trained policies, comparing them with previous SoTA methods and baseline policies. 'Policy x' refers to the policy optimized using the reward function 'RF x'. The 'RF x' column details the cumulative rewards for each policy, calculated in accordance with 'RF x'. Details of each reward function can be found in Table 6.1. The best value is highlighted in **bold**.

Florida Case	N Input (kg/ha) ↓	Irrigation (L/m^2) ↓	Yield (kg/ha) ↑	RF1 ↑	RF2 ↑	RF3 ↑	RF4 ↑	RF5 ↑
Empirical Baseline	360	394	10772	984	1417	1269	700	338
Policy1: Traditional Agent	200	**120**	10852	1425	1557	1538	1267	1673
Policy1: LM-based Agent (Ours)	**122**	192	**11402**	**1464**	**1675**	**1590**	**1337**	**1748**
Policy2: Traditional Agent	200	732	11244	813	1619	971	655	1020
Policy2: LM-based Agent (Ours)	**160**	**510**	**11474**	**1126**	**1687**	**1252**	**999**	**1330**
Policy3: Traditional Agent	19920	**108**	10865	-1.4e4	-1.4e4	1598	-3.0e4	-4.9e4
Policy3: LM-based Agent (Ours)	**10000**	264	**13152**	**-6.1e3**	**-5.8e3**	**1788**	**-1.4e4**	**-3.8e4**
Policy4: Traditional Agent	160	102	**10358**	1398	**1510**	1524	1272	1635
Policy4: LM-based Agent (Ours)	**160**	**36**	10192	**1428**	1468	**1555**	**1302**	**1647**
Policy5: Traditional Agent	200	138	10926	1417	1568	1575	1259	1651
Policy5: LM-based Agent (Ours)	**160**	**60**	**11280**	**1590**	**1656**	**1716**	**1463**	**1841**

6.3.3 Results of Experiments

The evaluation outcomes for the trained policies in both the Florida and Zaragoza case studies are detailed in Table 6.2, Table 6.3, and Figure 6.2. It's important to note that these results may not entirely reflect the optimal potential of the policies due to the random initialization of the Q-network and its

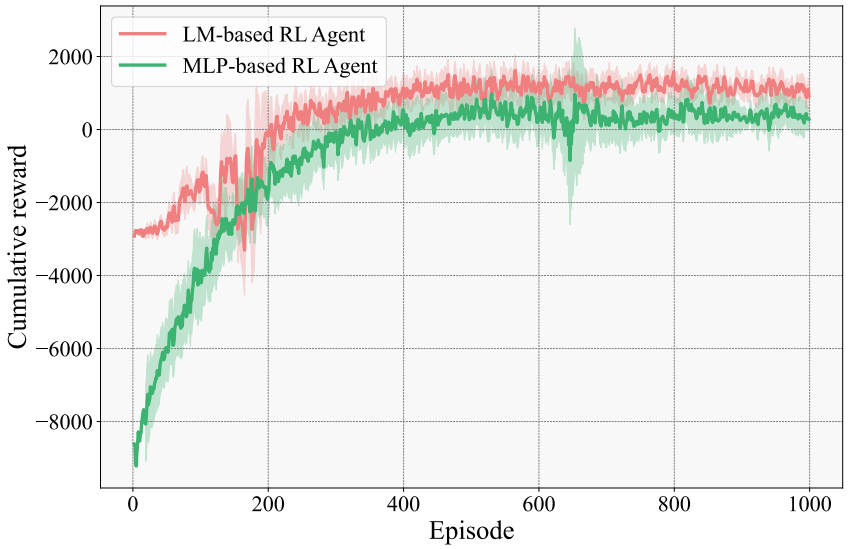

Figure 6.2: Cumulative reward versus episodes for policy training under RF1

episodic updates. Additionally, further refinement through hyperparameter tuning might yield more competitive outcomes. However, such tuning was intentionally avoided in this study to maintain a focus on generalizability and fair evaluation. Despite these deliverable-introduced constraints, the chosen policies still illustrate the effectiveness of the LM-based RL agent in enhancing crop management strategies. These policies also effectively demonstrate how different RFs can influence training outcomes.

The evaluation results, as detailed in Table 6.2 and Table 6.3, indicate that the proposed LM-based RL agent outperforms previous SoTA and empirical baselines in most metrics and scenarios. Notably, the LM-based RL agent consistently utilizes lower amounts of nitrogen and generally requires less irrigation, yet it manages to secure higher yields. These improvements are consistent across various reward functions that prioritize different optimization objectives, underscoring the agent's adaptability and robustness in optimizing for diverse agricultural goals. The findings validate the previous hypothesis that language models have a heightened capacity to decipher complex crop management scenarios and simulate environments, ultimately leading to the discovery of more optimal management practices. Compared with the baseline

policies, the RL-trained policies achieve a 49% and a 67% increase in terms of profit, i.e., RF1, and almost a 445% and a 37% increase in terms of RF5 for the Florida case and Zaragoza case, respectively. Notably, the enormous negative values of the cumulative rewards of Trained Policy 3 from both case studies are the results of the large amounts of N input, which are not punished during training with RF 3.

Consistent with prior studies [86], the choice of reward function significantly influences the strategy of policies trained with LM-based RL agents. For instance, when trained with RF2, which posits irrigation water as a free resource, Trained Policy 2 tends to maximize irrigation while minimizing nitrogen input. This approach leads to the highest yield and cumulative reward as per the criteria of RF2. In contrast, RF3 assumes zero cost for nitrogen fertilizer, prompting Trained Policy 3 to favor high nitrogen use and minimal irrigation in both case studies. Under RF4, which reflects a doubled cost of nitrogen fertilizer in comparison to RF1, Trained Policy 4 leads to a reduction in nitrogen use. Despite the reduced nitrogen input, this policy still achieves a substantial yield and notably saves over 64% of water resources, indicating the agent's capability to find a balance between cost efficiency and agricultural output.

In general, the results presented showcase the state-of-the-art capabilities of the LM-RL framework in optimizing crop management. This optimization is proven to be effective under various criteria, across different geographic locations, and within diverse environmental conditions. The framework's adaptability is highlighted by its ability to consistently apply LM-RL training to discover optimal management policies that align with specific targets, as dictated by the design of the chosen reward function. This flexibility and effectiveness affirm the potential of LM-RL as a powerful tool for agricultural management and decision-making.

Table 6.3: The evaluation results of our trained policies, comparing them with previous SoTA methods and baseline policies. 'Policy x' refers to the policy optimized using the reward function 'RF x'. The 'RF x' column details the cumulative rewards for each policy, calculated in accordance with 'RF x'. Details of each reward function can be found in Table 6.1. The best value is highlighted in **bold**.

Zaragoza Case	N Input (kg/ha) ↓	Irrigation (L/m^2) ↓	Yield (kg/ha) ↑	RF1 ↑	RF2 ↑	RF3 ↑	RF4 ↑	RF5 ↑
Empirical Baseline	250	752	10990	712	1539	909	514	1176
Policy1: Traditional Agent	240	330	10477	1103	1466	1292	913	1525
Policy1: LM-based Agent (Ours)	**160**	354	**10806**	**1192**	**1581**	**1318**	**1065**	**1617**
Policy2: Traditional Agent	200	1068	10923	393	1568	551	235	888
Policy2: LM-based Agent (Ours)	**160**	**1032**	**10856**	**453**	**1588**	**580**	**327**	**964**
Policy3: Traditional Agent	10640	324	10626	-7083	-6727	1323	-1.5e4	-8839
Policy3: LM-based Agent (Ours)	**10000**	342	**10903**	**-6553**	**-6177**	1347	**-8161**	**-8161**
Policy4: Traditional Agent	120	336	9601	1053	1422	1147	958	1464
Policy4: LM-based Agent (Ours)	**160**	**348**	**10250**	**1110**	**1493**	**1268**	**984**	**1542**
Policy5: Traditional Agent	200	390	10589	1086	1515	1244	928	1528
Policy5: LM-based Agent (Ours)	**160**	**362**	**10660**	**1160**	**1558**	**1286**	**1033**	**1610**

6.3.4 Ablation Studies

Training Separately on Fertilization and Irrigation

In our previous research endeavors, we concurrently optimized N fertilization and irrigation practices, subsequently comparing these results against both established baseline practices and previous SoTAs. To further elucidate the efficacy of this joint optimization approach, this section introduces an ablation study wherein the management policies for N fertilization and irrigation were trained independently. Specifically, while one practice was subject to optimization, the other adhered to established baseline methods. For instance, when optimizing an N management policy, the irrigation management followed the predefined baseline protocol, and vice versa. To be specific, experiments were conducted within the framework of the Florida case study, utilizing RF1 to guide the optimization process. The results, delineated in Table 6.4, provide a clear indication of the advantages inherent in the simultaneous optimization of N fertilization and irrigation management, as opposed to the independent optimization of each practice. This finding reveals that synergistically managing nitrogen fertilization and irrigation together yields superior agricultural

Fertilization	Irrigation	RF1↑
Baseline N Fertilization	Baseline Irrigation	984
Baseline N Fertilization	Training Irrigation	1376
Training N Fertilization	Baseline Irrigation	1157
Training N Fertilization	Training Irrigation	**1464**

Table 6.4: Performance comparison of the trained policies on both N fertilization and irrigation with the trained policies on either N fertilization or irrigation. The best values are shown in **bold**.

outcomes compared to optimizing each practice in isolation.

Exploration of Framework

In order to investigate the most effective framework of RL agents, an ablation study was conducted. This study aimed to ascertain the impact of the framework's structure on management practices. Aligning with the setup of our previous experiments, we present the results for the Florida case using the reward function RF1. The outcomes, as depicted in Table 6.5, indicate that employing a three-layer MLP yields the best results with a traditional RL agent. However, a notable decline in performance is observed when scaling the agent size from an MLP to a ResNet152 [22]. This performance drop suggests the occurrence of overfitting within the RL framework, implying that simply increasing the size of the neural network does not necessarily enhance the exploration of optimal management practices.

Contrastingly, the use of LMs, such as Distilled Bert, demonstrated a different trend. Not only did the LM exhibit improved performance, but it also provided valuable insights. The results suggest that LMs possess a unique ability to comprehend the underlying patterns and logic of crop and environmental models. This capability enables them to pinpoint more optimal solutions while successfully circumventing the issue of overfitting, which was observed with larger neural network models.

6.4 Path to Deployment

The effectiveness of management policies trained within the DSSAT-simulated environment may not directly translate to real-world scenarios. This potential discrepancy arises from uncertainties in weather conditions and differences between the crop models used for training and actual agricultural systems. This phenomenon, known as the *sim-to-real gap* [185], highlights a common challenge in applying RL policies, developed and refined in simulated settings, to practical, real-world environments.

6.4.1 Closing the Sim-To-Real Gap

To enhance the robustness of our trained management policies against the challenges posed by the *sim-to-real gap*, we plan to incorporate *domain and dynamics randomization* techniques, as suggested in previous studies [186], [187]. This approach involves introducing variations in critical parameters of the model and randomizing weather conditions during policy training. Such perturbations are intended to "force" the policies to become resilient to uncertainties in both the model and weather conditions.

While the primary focus of our current work is to establish the LM-based RL framework for crop management and to assess its effectiveness, we acknowledge the importance of addressing the robustness of these policies in real-world scenarios. Therefore, we aim to delve into this aspect in a forthcoming study, which will specifically target and evaluate the robustness of our LM-based RL policies against real-world variabilities and uncertainties.

6.4.2 Policy Evaluation with Measurement Noises

In order to assess the robustness of our method against random measurement noises, we conducted experiments following previous work [86]. In practical scenarios, farmers rely on weather forecasts and soil

Model Architecture	# of Parameters	RF1↑
Three-layer MLP	0.2M	1425
Five-layer MLP	0.5M	1312
ResNet18	11.0M	510
ResNet50	25.6M	230
ResNet101	44.7M	107
ResNet152	60.4M	110
Distilled Bert	60.3M	**1464**

Table 6.5: Performance comparison of different frameworks as RL agents. The best values are shown in **bold**.

moisture measurements to make informed decisions. However, these data sources often contain inaccuracies due to forecast errors and sensor limitations. To simulate this real-world scenario, we tested LM-based RL under policy 1 from the Florida case study by introducing random measurement noises to key observable state variables each day in the simulation. These noise values were determined based on the real-world accuracy data of weather forecasts and commonly used soil moisture meters [188]–[191]. For each variable of added noise, the policy's performance was evaluated 400 times, with the average cumulative reward and standard deviation reported. The results, detailed in Table 6.6, indicate that temperature and rainfall data inaccuracies have the most significant impact on policy performance, while other variables have minimal effects. Such an observation is consistent with previous research [86]. Notably, even with accumulated noise with multiple variables, the trained policy managed to achieve an average cumulative reward of 1248.8. While 15.3% lower than the reward in a noise-free environment, it is still considerably higher than that of the baseline policy. These findings demonstrate that the policies trained using our method can yield relatively satisfactory and robust results compared to baseline approaches, even under real-world scenarios.

Variables	Noises	RF 1	STD	Decrease (%)
Empirical Baseline	N/A	984.4	N/A	N/A
No Noise	N/A	1463.9	N/A	N/A
Soil water content	-+0.02	1463.9	0.0	0.0
Soil water content	-+0.05	1462.2	1.9	0.1
Temperature	-+1	1443.7	89.4	1.3
Temperature	-+2	1289.0	361.0	11.9
Solar Radiation	-+2%	1468.5	0.7	0
Solar Radiation	-+10%	1468.8	7.6	0
Rain Fall	90 % Acc.	1416.5	220.7	3.2
Leaf Area Index	-+10%	1457.1	1.2	0.4
Leaf Area Index	-+20%	1451.8	5.8	0.8
Soil water content	-+0.02			
+Temperature	-+2			
+Solar Radiation	-+2%	1248.8	386.8	15.3
+ Rain Fall	90 % Acc.			
+ Leaf Area Index	-+20%			

Table 6.6: Performance of the LM-based RL with Policy1 under measurement noises evaluated with RF1. The decrease (%) is calculated with respect to RF1, where no noise was applied.

Chapter 7

Discussion, Conclusion and Future Research

7.1 Discussion in Broader Domains

This dissertation is closely related to research in visual representation, agriculture, remote sensing, and earth observation. Meanwhile, it may also illuminate broader research domains, such as the efficient training of Generative Adversarial Networks(GANs). Brief related discussions on these topics will be presented in the following sections.

7.1.1 Efficient Training of Unsupervised Learning Algorithms

GANs [16], [192]–[194] are a type of generative model that has gained significant attention in recent years due to their impressive performance in image-generation tasks. However, the mainstream models in

GANs are known to be computationally intensive, making them challenging to train in resource-constrained settings. Therefore, it is crucial to develop methods that can effectively reduce the computational cost of training GANs while maintaining their performance, making GANs more practical and applicable in real-world scenarios.

Neural network pruning has recently emerged as a powerful tool to reduce the training and inference costs of DNNs for supervised learning [195]. There are three main genres of pruning methods: pruning-at-initialization, pruning-during-training, and post-hoc pruning methods. Post-hoc pruning [196]–[198] can date back to the 1980s, which was first introduced for reducing inference time and memory requirements for efficient deployment; hence does not align with our purpose of efficient training. Later, pruning-at-initialization [199]–[201] and pruning-during-training methods [202] were introduced to circumvent the need to fully train the dense networks. However, early pruning-during-training algorithms [203] do not bring much training efficiency compared to post-hoc pruning, while pruning-at-initialization methods usually suffer from significant performance drop [204]. Recently, advances in dynamic sparse training (DST) [205]–[209] for the first time show that pruning-during-training methods can have comparable training FLOPs as pruning-at-initialization methods while having competing performance to post-hoc pruning. Therefore, applying DST on GANs seems to be a promising choice.

Although DST has attained remarkable achievements in supervised learning, the application of DST on GANs is not successful due to newly emerging challenges. One challenge is keeping the generator and the discriminator balanced. In particular, using overly strong discriminators can lead to overfitting, while weaker discriminators may fail to effectively prevent mode collapse [210], [211]. Hence, balancing the sparse generator and the (possibly) sparse discriminator throughout training is even more difficult. To mitigate the imbalance issue, a recent work STU-GAN [212] proposes to apply DST directly to the

generator. However, we find empirically that such an algorithm is likely to fail when the generator is already more powerful than the discriminator. Consequently, it remains unclear how to conduct balanced dynamic sparse training for GANs.

To this end, we propose a metric called balance ratio (BR), which measures the degree of balance of the two components, to study sparse GAN training. We find that BR is useful in (1) understanding the interaction between the discriminator and the generator, (2) identifying the cause of a certain training failure/collapse [192], [213], and (3) helping stabilize sparse GAN training as an indicator. To our best knowledge, this is the first study to quantify the imbalance of sparse GANs and may even provide new insights into dense GAN training.

7.2 Conclusion and Further Research

This dissertation explored the nature of representation learning and possible techniques to improve its quality. Additionally, we investigate representation learning's ability and performance in three essential and under-explored areas: agriculture, remote sensing, and earth observation.

In Chapter 2, we propose *Hallucinator*, which generates additional hard positive pairs for contrastive learning models based on Siamese structure. *Hallucinator* generates novel data samples in the feature space to provide the training with further contrast without additional computation. We design an asymmetric feature extrapolation to avoid trivial positive pairs and innovatively introduce non-linear hallucination to smooth the generated samples. We empirically prove the effectiveness and generalization capacity of *Hallucinator* to well-recognized contrastive learning models, including MoCoV1&V2, SimCLR, and SimSiam. Finally, we hope this work could bring the concept of "Hallucination" into the SSL domain and unlock future research on sample generations&synthesis in contrastive learning across different areas and

diverse modalities.

In Chapter 3, we step into the application of representation learning and its related datasets in agriculture and remote sensing. Large, high-quality datasets are opening tremendous new opportunities for computational agriculture, but they are extremely difficult to obtain. As in other domains, remote sensing and earth observation data are marked by huge amounts of unlabeled data and relatively few annotations; leveraging the information in this unlabeled data, therefore, becomes a critical task. In this work, we contribute to the advancement of these efforts by releasing the AV+ dataset, which contains annotated full-field imagery based on the original AV dataset [27], supplemented by more than 3TB of raw full-field images taken at different times in the season. The improved supervised component of the AV dataset will allow for greater flexibility in training and augmentation protocols and enable additional possible lines of study around long-range context and large-scale imagery. The raw unlabeled data will enable continued exploration in the self, semi, and weakly supervised methods which we have begun to benchmark here. This extension of an already important dataset in computational agriculture will open up many lines of research and investigation which benefit both the agriculture and computer vision communities. Next, we conduct a thorough benchmark study on self-supervised pre-training methods based on contrastive learning, which captures the fine-grained, spatiotemporal nature of this data. We analyze a classification formulation of the AV+ dataset under linear probing, non-linear probing, and fine-tuning. We also examined segmentation tasks, which are often overlooked in remote sensing approaches, based on the original segmentation formulation of AV+ with a frozen and unfrozen encoder and an extremely small fine-grained segmentation task under the same formulations. Our benchmark study explores both traditional CNN architectures (ResNet-18 and ResNet-50) as well as the more recent Swin Transformer, which offers unique potentials for computer vision, but requires huge amounts of data to train. Importantly,

we incorporate the Pixel-to-Propagation Module, originally built in the SimCLR framework, into the MoCo-V2 framework, which allows for training on larger batch sizes. Our results show that this module is key for downstream segmentation and *classification* tasks, even though it was designed primarily for dense detection and segmentation tasks. As our dataset contains richer low-level, high-frequency, fine-grained features than traditional natural imagery like COCO or ImageNet, this suggests that PPM is beneficial for learning dense, fine-grained *features* in addition to dense label structure. We further combine this module with a TemCo, a modification of SeCo, into a rich framework that captures the dense, spatiotemporal structure of our data. While this combined framework was not the highest-performing on the various task, it again may have been at a disadvantage since it is a larger model and the number of steps was fixed for a fair comparison. Additionally, extending how *positive* samples are generated could prove beneficial. These improvements are the focus of future analysis. Self-supervised methods will be crucial for unlocking opportunities in remote sensing, particularly for agriculture, and this dataset release and benchmark study offer a significant step in that direction.

In Chapter 4, we continue the discussion of representation in few-shot learning settings in the remote sensing and earth observation areas. We propose GenCo, a generator-based two-stage approach for few-shot classification and segmentation on remote sensing and earth observation data. Our method proved to be effective due to the sufficient contrast introduced by the generator during pre-training and the remarkable adaptability of embeddings in downstream few-shot tasks. Furthermore, we demonstrated that the generator could be integrated into the few-shot learning framework to further address the issue of data scarcity and enrich the learning information. Most importantly, our approach provides an alternative solution to the labeling challenges in agriculture and remote sensing domains. With our few-shot contrastive learning-based approach, we believe that it is possible to deploy models in the real world with minimal

labeled data and training effort.

In Chapter 5, we present a framework for optimizing N management with deep RL and crop simulations based on DSSAT. With the proposed framework, we train management policies with deep Q-network (DQN) and soft actor-critic (SAC) for the maize crop in Iowa and Florida, which are shown to outperform standard management practices. We also evaluate the effect of partial observation and reduced action frequencies. We believe our work demonstrates the potential of deep RL in optimizing crop management for more sustainable and resilient agriculture.

In Chapter 6, We address the crucial challenge of optimizing crop management to maximize yield while minimizing management costs and environmental impacts. We present an innovative framework that combines deep reinforcement learning, language models, and crop simulations using Gym-DSSAT. The experimental results clearly demonstrate that Language Model-based Reinforcement Learning agents surpass baseline models and significantly outperform existing SoTA methods. This enhanced performance stems from the LM-RL agents' capacity to dynamically adjust their strategies according to different reward function designs, coupled with their ability to think and infer like expert agronomists. This dual capability enables them to maximize rewards in a variety of scenarios. Crucially, the framework has proven effective even in the presence of measurement noise in observable state variables, which is particularly promising for real-world applications. We aspire for our work to serve as a proof of concept for the potential of LMs as adept agronomists, sparking interest and motivating further exploration in this area. The ultimate goal is to encourage researchers and practitioners to investigate and implement more advanced language models in practical agricultural settings. We believe that such advancements could significantly contribute to the evolution of agricultural technology, leading to smarter, more efficient, and sustainable farming practices worldwide.

References

[1] J.-B. Grill, F. Strub, F. Altché, *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[2] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, PMLR, 2020, pp. 1597–1607.

[4] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.

[5] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?" *Advances in neural information processing systems*, vol. 33, pp. 6827–6839, 2020.

[6] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[7] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 750–15 758.

[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–308, 2009.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[10] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[11] P. Chen, S. Liu, and J. Jia, "Jigsaw clustering for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 526–11 535.

[12] X. Peng, K. Wang, Z. Zhu, M. Wang, and Y. You, "Crafting better contrastive views for siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 031–16 040.

[13] Z. Shen, Z. Liu, Z. Liu, M. Savvides, T. Darrell, and E. Xing, "Un-mix: Rethinking image mixtures for unsupervised visual representation learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 2216–2224.

[14] R. Zhu, B. Zhao, J. Liu, Z. Sun, and C. W. Chen, "Improving contrastive learning by visualizing feature transformation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 306–10 315.

[15] M. D. Grilli and E. L. Glisky, "The self-imagination effect: Benefits of a self-referential encoding strategy on cued recall in memory-impaired individuals with neurological damage," *Journal of the International Neuropsychological Society*, vol. 17, no. 5, pp. 929–933, 2011.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[17] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[19] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[20] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7278–7286.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[25] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[26] O. Mañas, A. Lacoste, X. Giro-i-Nieto, D. Vazquez, and P. Rodriguez, "Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data," *arXiv preprint arXiv:2103.16607*, 2021.

[27] M. T. Chiu, X. Xu, Y. Wei, *et al.*, "Agriculture-vision: A large aerial image database for agricultural pattern analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2828–2838.

[28] Y. Wang, J. Wu, N. Hovakimyan, and R. Sun, "Balanced training for sparse gans," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[29] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, 2016.

[30] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*, PMLR, 2017, pp. 1126–1135.

[31] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[32] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4367–4375.

[33] X. Sun, B. Wang, Z. Wang, H. Li, H. Li, and K. Fu, "Research progress on few-shot learning for remote sensing image interpretation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2387–2402, 2021.

[34] D. Alajaji, H. S. Alhichri, N. Ammour, and N. Alajlan, "Few-shot learning for remote sensing scene classification," in *2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS)*, IEEE, 2020, pp. 81–84.

[35] X. Li, J. Deng, and Y. Fang, "Few-shot object detection on remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[36] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *International Conference on Machine Learning*, PMLR, 2021, pp. 12 310–12 320.

[37] J. Wu, D. Pichler, D. Marley, D. Wilson, N. Hovakimyan, and J. Hobbs, "Extended agriculture-vision: An extension of a large aerial image dataset for agricultural pattern analysis," *arXiv preprint arXiv:2303.02460*, 2023.

[38] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5822–5830.

[39] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu, "Frustratingly simple few-shot object detection," *arXiv preprint arXiv:2003.06957*, 2020.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[41] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?" In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.

[42] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML workshop on unsupervised and transfer learning*, JMLR Workshop and Conference Proceedings, 2012, pp. 17–36.

[43] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[45] S. Chen, J. Wu, N. Hovakimyan, and H. Yao, "Recontab: Regularized contrastive representation learning for tabular data," *arXiv preprint arXiv:2310.18541*, 2023.

[46] J. Wu, J. Hobbs, and N. Hovakimyan, "Hallucination improves the performance of unsupervised visual representation learning," *arXiv preprint arXiv:2307.12168*, 2023.

[47] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in neural information processing systems*, vol. 32, 2019.

[48] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *International conference on machine learning*, PMLR, 2020, pp. 4182–4192.

[49] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," *arXiv preprint arXiv:2005.04966*, 2020.

[50] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, IEEE, vol. 2, 2006, pp. 1735–1742.

[51] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, "Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 684–16 693.

[52] M. Cordts, M. Omran, S. Ramos, *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[53] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[54] A. Gupta, P. Dollar, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5356–5364.

[55] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 633–641.

[56] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, 2017, pp. 3226–3229.

[57] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, 2019.

[58] I. Demir, K. Koperski, D. Lindenbaum, *et al.*, "Deepglobe 2018: A challenge to parse the earth through satellite images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 172–181.

[59] R. C. Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Urban change detection for multispectral earth observation using convolutional neural networks," *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2115–2118, 2018.

[60] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "Bigearthnet: A large-scale benchmark archive for remote sensing image understanding," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 5901–5904, 2019. DOI: 10.1109/IGARSS.2019.8900532.

[61] G. Tseng, I. Zvonkov, C. L. Nakalembe, and H. Kerner, "Cropharvest: A global dataset for crop-type classification," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[62] M. Feng and Y. Bai, "A global land cover map produced through integrating multi-source datasets," *Big Earth Data*, vol. 3, no. 3, pp. 191–219, 2019.

[63] S. Haug and J. Ostermann, "A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks," 2014.

[64] A. Olsen, D. A. Konovalov, B. Philippa, *et al.*, "Deepweeds: A multiclass weed species image dataset for deep learning," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[65] M. T. Chiu, X. Xu, K. Wang, *et al.*, "The 1st agriculture-vision challenge: Methods and results," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 48–49.

[66] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3018–3027.

[67] E. Schwartz, L. Karlinsky, J. Shtok, *et al.*, "Delta-encoder: An effective sample synthesis method for few-shot object recognition," *Advances in neural information processing systems*, vol. 31, 2018.

[68] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan: An adversarial approach to few-shot learning," *Advances in neural information processing systems*, vol. 31, 2018.

[69] W. Zhang and Y.-X. Wang, "Hallucination improves few-shot object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 008–13 017.

[70] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li, "Attention-aware face hallucination via deep reinforcement learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 690–698.

[71] H. Zhang, J. Zhang, and P. Koniusz, "Few-shot learning via saliency-guided hallucination of samples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2770–2779.

[72] L. Gui, A. Bardes, R. Salakhutdinov, A. Hauptmann, M. Hebert, and Y.-X. Wang, "Learning to hallucinate examples from extrinsic and intrinsic supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8701–8711.

[73] A. Shah, A. Roy, K. Shah, *et al.*, "Halp: Hallucinating latent positives for skeleton-based self-supervised learning of actions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 846–18 856.

[74] Z. Bao, Y.-X. Wang, and M. Hebert, "Bowtie networks: Generative modeling for joint few-shot recognition and novel-view synthesis," *arXiv preprint arXiv:2008.06981*, 2020.

[75] K. Gong, B. Li, J. Zhang, *et al.*, "Posetriplet: Co-evolving 3d human pose estimation, imitation, and hallucination under self-supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 017–11 027.

[76] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," *arXiv preprint arXiv:1702.05538*, 2017.

[77] V. Kumar, H. Glaude, C. de Lichy, and W. Campbell, "A closer look at feature space data augmentation for few-shot intent classification," *arXiv preprint arXiv:1910.04176*, 2019.

[78] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu, "Consert: A contrastive framework for self-supervised sentence representation transfer," *arXiv preprint arXiv:2105.11741*, 2021.

[79] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," *arXiv preprint arXiv:2104.08821*, 2021.

[80] J. Wu, N. Hovakimyan, and J. Hobbs, "Genco: An auxiliary generator from contrastive learning for enhanced few-shot learning in remote sensing," *arXiv preprint arXiv:2307.14612*, 2023.

[81] J. Li, W. Qiang, C. Zheng, B. Su, and H. Xiong, "Metaug: Contrastive learning via meta feature augmentation," in *International Conference on Machine Learning*, PMLR, 2022, pp. 12 964–12 978.

[82] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5058–5066.

[83] Y. Wang, J. Wu, N. Hovakimyan, and R. Sun, "Double dynamic sparse training for gans," *arXiv preprint arXiv:2302.14670*, 2023.

[84] H. Wang, Y. Wang, R. Sun, and B. Li, "Global convergence of maml and theory-inspired neural architecture search for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9797–9808.

[85] J. Wu, R. Tao, P. Zhao, N. F. Martin, and N. Hovakimyan, "Optimizing nitrogen management with deep reinforcement learning and crop simulations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1712–1720.

[86] R. Tao, P. Zhao, J. Wu, *et al.*, "Optimizing crop management with reinforcement learning and imitation learning," *arXiv preprint arXiv:2209.09991*, 2022.

[87] X. Chen, L. Yao, T. Zhou, J. Dong, and Y. Zhang, "Momentum contrastive learning for few-shot covid-19 diagnosis from chest ct images," *Pattern recognition*, vol. 113, p. 107 826, 2021.

[88] S. Chen, N. Kong, X. Sun, H. Meng, and M. Li, "Claims data-driven modeling of hospital time-to-readmission risk with latent heterogeneity," *Health care management science*, vol. 22, pp. 156–179, 2019.

[89] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[90] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9197–9206.

[91] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *Conference on robot learning*, PMLR, 2017, pp. 357–368.

[92] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, No specific page numbers, Lille, vol. 2, 2015.

[93] Y. L. Cacheux, H. L. Borgne, and M. Crucianu, "Modeling inter and intra-class relations in the triplet loss for zero-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10 333–10 342.

[94] S. Gidaris and N. Komodakis, "Generating classification weights with gnn denoising autoencoders for few-shot learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 21–30.

[95] A. Nichol and J. Schulman, "Reptile: A scalable metalearning algorithm," *arXiv preprint arXiv:1803.02999*, vol. 2, no. 3, p. 4, 2018.

[96] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 657–10 665.

[97] K. Li, Y. Zhang, K. Li, and Y. Fu, "Adversarial feature hallucination networks for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 470–13 479.

[98] L. Li, J. Han, X. Yao, G. Cheng, and L. Guo, "Dla-matchnet for few-shot remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 7844–7853, 2020.

[99] D. A. Alajaji and H. Alhichri, "Few shot scene classification in remote sensing using meta-agnostic machine," in *2020 6th conference on data science and machine learning applications (CDMA)*, IEEE, 2020, pp. 77–80.

[100] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2290–2304, 2018.

[101] B. Wang, Z. Wang, X. Sun, H. Wang, and K. Fu, "Dmml-net: Deep metametric learning for few-shot geographic object segmentation in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2021.

[102] R. Kemker, R. Luu, and C. Kanan, "Low-shot learning for the semantic segmentation of remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, pp. 6214–6223, 2018.

[103] X. Yao, Q. Cao, X. Feng, G. Cheng, and J. Han, "Scale-aware detailed matching for few-shot aerial image semantic segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2021.

[104] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, "Boosting few-shot visual learning with self-supervision," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8059–8068.

[105] J.-C. Su, S. Maji, and B. Hariharan, "When does self-supervision improve few-shot learning?" In *European conference on computer vision*, Springer, 2020, pp. 645–666.

[106] J. Wu, S. Chen, Q. Zhao, *et al.*, "Switchtab: Switched autoencoders are effective tabular learners," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 15 924–15 933.

[107] C. Doersch, A. Gupta, and A. Zisserman, "Crosstransformers: Spatially-aware few-shot transfer," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 981–21 993, 2020.

[108] Q. Zeng and J. Geng, "Task-specific contrastive learning for few-shot remote sensing image scene classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 191, pp. 143–154, 2022.

[109] H. Li, Y. Li, G. Zhang, *et al.*, "Global and local contrastive self-supervised learning for semantic segmentation of hr remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[110] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning*, PMLR, 2022, pp. 9118–9147.

[111] W. Huang, F. Xia, T. Xiao, *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[112] S. S. Raman, V. Cohen, E. Rosen, I. Idrees, D. Paulius, and S. Tellex, "Planning with large language models via corrective re-prompting," in *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[113] O. Mees, J. Borja-Diaz, and W. Burgard, "Grounding language with visual affordances over unstructured data," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 11 576–11 582.

[114] B. Chen, F. Xia, B. Ichter, *et al.*, "Open-vocabulary queryable scene representations for real world planning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 11 509–11 522.

[115] M. Ahn, A. Brohan, N. Brown, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[116] Y. Wang, J. Su, H. Lu, *et al.*, "Lemon: Lossless model expansion," *arXiv preprint arXiv:2310.07999*, 2023.

[117] J. Liang, W. Huang, F. Xia, *et al.*, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 9493–9500.

[118] Z. Lai, J. Wu, S. Chen, Y. Zhou, A. Hovakimyan, and N. Hovakimyan, "Language models are free boosters for biomedical imaging tasks," *arXiv preprint arXiv:2403.17343*, 2024.

[119] J. Wu, Z. Lai, S. Chen, R. Tao, P. Zhao, and N. Hovakimyan, "The new agronomists: Language models are experts in crop management," *arXiv preprint arXiv:2403.19839*, 2024.

[120] S. Liu, J. Wu, J. Bao, W. Wang, N. Hovakimyan, and C. G. Healey, "Towards a robust retrieval-based summarization system," *arXiv preprint arXiv:2403.19889*, 2024.

[121] G. Kim, P. Baldi, and S. McAleer, "Language models can solve computer tasks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[122] S. Yao, J. Zhao, D. Yu, *et al.*, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022.

[123] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[124] F. Garcia, "Use of reinforcement learning and simulation to optimize wheat crop technical management," in *Proceedings of the International Congress on Modelling and Simulation*, 1999, pp. 801–806.

[125] L. Sun, Y. Yang, J. Hu, D. Porter, T. Marek, and C. Hillyer, "Reinforcement learning control for water-efficient agricultural irrigation," in *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, 2017, pp. 1334–1341.

[126] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[127] O. Vinyals, I. Babuschkin, W. M. Czarnecki, *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[128] C. Gamble and J. Gao, *Safety-first AI for autonomous data centre cooling and industrial control*. [Online]. Available: https://deepmind.com/blog/article/safety-first-ai-autonomous-data-centre-cooling-and-industrial-control.

[129] E. Kaufmann, A. Loquercio, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, "Deep drone racing: Learning agile flight in dynamic environments," in *Conference on Robot Learning*, 2018, pp. 133–145.

[130] J. Hwangbo, J. Lee, A. Dosovitskiy, *et al.*, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.

[131] Y. Song, M. Steinweg, E. Kaufmann, and D. Scaramuzza, "Autonomous drone racing with deep reinforcement learning," *arXiv preprint arXiv:2103.08624*, 2021.

[132] H. Overweg, H. N. Berghuijs, and I. N. Athanasiadis, "CropGym: A reinforcement learning environment for crop management," *arXiv preprint arXiv:2104.04326*, 2021.

[133] C. Ashcraft and K. Karra, "Machine learning aided crop yield optimization," *arXiv preprint arXiv:2111.00963*, 2021.

[134] J. W. Jones, J. M. Antle, B. Basso, *et al.*, "Brief history of agricultural systems modeling," *Agricultural systems*, vol. 155, pp. 240–254, 2017.

[135] T. J. Salo, T. Palosuo, K. C. Kersebaum, *et al.*, *The Journal of Agricultural Science*, vol. 154, no. 7, pp. 1218–1240, 2016.

[136] G. Brockman, V. Cheung, L. Pettersson, *et al.*, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[137] M. Shibu, P. Leffelaar, H. Van Keulen, and P. Aggarwal, "LINTUL3, a simulation model for nitrogen-limited situations: Application to rice," *European Journal of Agronomy*, vol. 32, no. 4, pp. 255–271, 2010.

[138] C. Zhao, B. Liu, L. Xiao, *et al.*, "A SIMPLE crop model," *European Journal of Agronomy*, vol. 104, pp. 97–106, 2019.

[139] R. Gautron and E. J. Padrón González, *gym-DSSAT - A crop model turned into a Reinforcement Learning environment*, Mar. 2022. [Online]. Available: https://gitlab.inria.fr/rgautron/gym_dssat_pdi.

[140] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[141] H. Guo, "Nonlinear mixup: Out-of-manifold data augmentation for text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 4044–4051.

[142] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," ., 2009.

[143] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning*, PMLR, 2020, pp. 9929–9939.

[144] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, *Detectron2*, https://github.com/facebookresearch/detectron2, 2019.

[145] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[146] Z. Liu, Y. Lin, Y. Cao, *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[147] A. Monteiro and A. von Wangenheim, *Orthomosaic dataset of rgb aerial images for weed mapping*, http://www.lapix.ufsc.br/weed-mapping-sugar-cane, 2019.

[148] S. Haug and J. Ostermann, "A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks," in *European conference on computer vision*, Springer, 2015, pp. 105–116.

[149] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.

[150] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[151] T. Han, W. Xie, and A. Zisserman, "Memory-augmented dense predictive coding for video representation learning," in *European conference on computer vision*, Springer, 2020, pp. 312–329.

[152] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[153] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[154] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 464–472. DOI: 10.1109/WACV.2017.58.

[155] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. arXiv: 1708.02002. [Online]. Available: http://arxiv.org/abs/1708.02002.

[156] M. Neumann, A. S. Pinto, X. Zhai, and N. Houlsby, "In-domain representation learning for remote sensing," *arXiv preprint arXiv:1911.06721*, 2019.

[157] M. T. Chiu, X. Xu, K. Wang, *et al.*, "The 1st agriculture-vision challenge: Methods and results," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2020.

[158] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in neural information processing systems*, vol. 33, pp. 22 243–22 255, 2020.

[159] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, International Society for Optics and Photonics, vol. 11006, 2019, p. 1 100 612.

[160] T. Searchinger, R. Waite, C. Hanson, *et al.*, *Creating a sustainable food future: A menu of solutions to feed nearly 10 billion people by 2050. Final report*. World Resources Institute, 2019.

[161] M. A. Sutton, O. Oenema, J. W. Erisman, A. Leip, H. van Grinsven, and W. Winiwarter, "Too much of a good thing," *Nature*, vol. 472, no. 7342, pp. 159–161, 2011.

[162] I. Akkaya, M. Andrychowicz, M. Chociej, *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[163] M. W. Palmer, J. Cooper, C. Tétard-Jones, *et al.*, "The influence of organic and conventional fertilisation and crop protection practices, preceding crop, harvest year and weather conditions on yield and quality of potato (Solanum tuberosum) in a long-term management trial," *European Journal of Agronomy*, vol. 49, pp. 83–92, 2013.

[164] J.-M. Attonaty, M.-H. Chatelin, F. Garcia, and S. Ndiaye, "Using extended machine learning and simulation technics to design crop management strategies," in *EFITA First European Conference for Information Technology in Agriculture, Copenhagen (Denmark)*, 1997.

[165] J.-E. Bergez, N. Colbach, O. Crespo, *et al.*, "Designing crop management systems by simulation," *European Journal of Agronomy*, vol. 32, no. 1, pp. 3–9, 2010.

[166] J. W. Jones, G. Hoogenboom, C. H. Porter, *et al.*, "The DSSAT cropping system model," *European Journal of Agronomy*, vol. 18, no. 3-4, pp. 235–265, 2003.

[167] G. Hoogenboom, K. B. C.H. Porter, V. Shelia, *et al.*, "The DSSAT crop modeling ecosystem," in *Advances in Crop Modeling for a Sustainable Agriculture*, K. Boote, Ed., Cambridge, United Kingdom: Burleigh Dodds Science Publishing, 2019, pp. 173–216.

[168] J. Jin, C. Song, H. Li, K. Gai, J. Wang, and W. Zhang, "Real-time bidding with multi-agent reinforcement learning in display advertising," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 2193–2201.

[169] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, PMLR, 2018, pp. 1861–1870.

[170] C. Richardson, "Weather simulation for crop management models," *Transactions of the ASAE*, vol. 28, no. 5, pp. 1602–1606, 1985.

[171] G. Mandrini, C. M. Pittelkow, S. V. Archontoulis, T. Mieno, and N. F. Martin, "Understanding differences between static and dynamic nitrogen fertilizer tools using simulation modeling," *Agricultural Systems*, vol. 194, p. 103 275, 2021.

[172] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[173] B. Reddy, P. S. Reddy, F. Bidinger, and M. Blümmel, "Crop management factors influencing yield and quality of crop residues," *Field Crops Research*, vol. 84, no. 1-2, pp. 57–77, 2003.

[174] D. Wright, I. Small, C. Mackowiak, Z. Grabau, P. Devkota, and S. Paula-Moraes, "Field corn production guide: Ss-agr-85/ag202, rev. 8/2022," *EDIS*, vol. 2022, no. 4, 2022.

[175] A. Skhiri and F. Dechmi, "Impact of sprinkler irrigation management on the del reguero river (spain). i: Water balance and irrigation performance," *Agricultural Water Management*, vol. 103, pp. 120–129, 2012.

[176] I. Ara, L. Turner, M. T. Harrison, M. Monjardino, P. DeVoil, and D. Rodriguez, "Application, adoption and opportunities for improving decision support systems in irrigated agriculture: A review," *Agricultural Water Management*, vol. 257, p. 107 161, 2021.

[177] G. Romain, P. Philippe, B. Julien, M. Odalric-Ambrym, E. David, *et al.*, "Gym-dssat: A crop model turned into a reinforcement learning environment," *arXiv preprint arXiv:2207.03270*, 2022.

[178] J. Wei, X. Wang, D. Schuurmans, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[179] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–22.

[180] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[181] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[182] W. Malik, R. Isla, and F. Dechmi, "Dssat-ceres-maize modelling to improve irrigation and nitrogen management practices under mediterranean conditions," *Agricultural Water Management*, vol. 213, pp. 298–308, 2019.

[183] G. Mandrini, C. M. Pittelkow, S. Archontoulis, D. Kanter, and N. F. Martin, "Exploring trade-offs between profit, yield, and the environmental footprint of potential nitrogen fertilizer regulations in the us midwest," *Frontiers in plant science*, vol. 13, 2022.

[184] H. Di and K. Cameron, "Nitrate leaching in temperate agroecosystems: Sources, factors and mitigating strategies," *Nutrient cycling in agroecosystems*, vol. 64, pp. 237–256, 2002.

[185] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: A survey," in *2020 IEEE symposium series on computational intelligence (SSCI)*, IEEE, 2020, pp. 737–744.

[186] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[187] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3803–3810.

[188] D. R. Cobos and D. Devices, "Why does my soil moisture sensor read negative?" *Online]. Available: http://manuals. decagon. com/RetiredandDiscontinued/Slicksandcontent/ Presentations/SoilMoisture301.pdf*, 2010.

[189] E. Floehr, "Weather forecast accuracy analysis," in *Proc of the 9th Python in Science Conference, SciPy*, 2010, pp. 36–39.

[190] P. Zhang, Y. Jia, J. Gao, W. Song, and H. Leung, "Short-term rainfall forecasting using multi-layer perceptron," *IEEE Transactions on Big Data*, vol. 6, no. 1, pp. 93–106, 2018.

[191] D. Heinemann, E. Lorenz, and M. Girodo, "Forecasting of solar radiation," *Solar energy resource management for electricity generation from local level to global scale. Nova Science Publishers, New York*, pp. 83–94, 2006.

[192] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[193] A. Sauer, K. Schwarz, and A. Geiger, "Stylegan-xl: Scaling stylegan to large diverse datasets," in *ACM SIGGRAPH 2022 conference proceedings*, 2022, pp. 1–10.

[194] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. Han, "Autoregressive image generation using residual quantization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 523–11 532.

[195] Y. Wang, D. Li, and R. Sun, "Ntk-sap: Improving neural network pruning by aligning training dynamics," *arXiv preprint arXiv:2304.02840*, 2023.

[196] S. A. Janowsky, "Pruning versus clipping in neural networks," *Physical Review A*, vol. 39, no. 12, p. 6600, 1989.

[197] Y. LeCun, J. Denker, and S. Solla, "Optimal brain damage," *Advances in neural information processing systems*, vol. 2, 1989.

[198] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.

[199] N. Lee, T. Ajanthan, and P. H. Torr, "Snip: Single-shot network pruning based on connection sensitivity," *arXiv preprint arXiv:1810.02340*, 2018.

[200] C. Wang, G. Zhang, and R. Grosse, "Picking winning tickets before training by preserving gradient flow," *arXiv preprint arXiv:2002.07376*, 2020.

[201] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6377–6389, 2020.

[202] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[203] C. Louizos, M. Welling, and D. P. Kingma, "Learning sparse neural networks through L_0 regularization," *arXiv preprint arXiv:1712.01312*, 2017.

[204] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin, "Pruning neural networks at initialization: Why are we missing the mark?" In *ICLR*, 2021.

[205] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta, "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science," *Nature communications*, vol. 9, no. 1, pp. 1–12, 2018.

[206] U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen, "Rigging the lottery: Making all tickets winners," in *International Conference on Machine Learning*, PMLR, 2020, pp. 2943–2952.

[207] S. Liu, T. Chen, X. Chen, *et al.*, "Sparse training via boosting pruning plasticity with neuroregeneration," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9908–9922, 2021.

[208] S. Liu, D. C. Mocanu, Y. Pei, and M. Pechenizkiy, "Selfish sparse rnn training," in *International Conference on Machine Learning*, PMLR, 2021, pp. 6893–6904.

[209] S. Liu, L. Yin, D. C. Mocanu, and M. Pechenizkiy, "Do we actually need dense over-parameterization? in-time over-parameterization in sparse training," in *International Conference on Machine Learning*, PMLR, 2021, pp. 6989–7000.

[210] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," in *International Conference on Machine Learning*, PMLR, 2017, pp. 224–232.

[211] Y. Bai, T. Ma, and A. Risteski, "Approximability of discriminators implies diversity in gans," *arXiv preprint arXiv:1806.10586*, 2018.

[212] S. Liu, Y. Tian, T. Chen, and L. Shen, "Don't be so dense: Sparse-to-sparse gan training without sacrificing performance," *arXiv preprint arXiv:2203.02770*, 2022.

[213] T. Chen, Y. Cheng, Z. Gan, J. Liu, and Z. Wang, "Data-efficient gan training beyond (just) augmentations: A lottery ticket perspective," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 941–20 955, 2021.

[214] Z. Y. Shang Liu Jianlong Yuan, *Grid concatenation for agriculture semantic segmentation*, https://www.youtube.com/watch?v=-X3XOzDhZAM&list=PLPtQK8rJZ9HyQM1XBOvY09OmnCsQlQRCz&index=1.

[215] J. W. Rouse, R. H. Haas, J. A. Schell, D. W. Deering, *et al.*, "Monitoring vegetation systems in the great plains with erts," *NASA special publication*, vol. 351, no. 1974, p. 309, 1974.

[216] J. Xue and B. Su, "Significant remote sensing vegetation indices: A review of developments and applications," *Journal of sensors*, vol. 2017, 2017.

Appendix A

Improved Representation Learning

A.1 Additional Model Pipelines

We illustrate how *Hallucinator* plugged into other models. To be specific, Figure A.1 demonstrates the overall pipeline of MoCoV1. The corresponding InfoNCE loss [150] could be defined as:

$$\mathcal{L}_{MoCo} = -\log \frac{\exp(q \cdot k^{/}/\tau) + \exp(\hat{q} \cdot k^{/}/\tau)}{\sum_{k^-} \exp(q \cdot k^-/\tau) + \exp(q \cdot k/\tau) + \exp(\hat{q} \cdot k^{/}/\tau)}, \tag{A.1}$$

where τ is a temperature hyper-parameter, (q, k) and $(\hat{q}, k)$ are two positive pairs. (q, k^-) are negative pairs. All k^- vectors are stored in a queue structure.

Following this, we demonstrate the pipeline of SimCLR [3] with the proposed *Hallucinator* in Figure A.2. Again, (q, k) and $(\hat{q}, k)$ are defined as two positive pairs for consistency. For a positive pair (q, k), we define the loss to be the same as before [3]:

$$\mathcal{L}_{SimCLR}(q,k) = -\log \frac{\exp(sim(q,k)/\tau)}{\sum_{k^-}^{4N} \mathbb{1}_{[k^- \neq q]} \exp(sim(q,k^-)/\tau)}, \tag{A.2}$$

where $sim(q,k) = q^T k / \|q\| \, \|k\|$, i.e., cosine similarity. N represents the batch size and τ is a temperature hyper-parameter. Originally, we had $2N$ data points. The number increases to $4N$ as the *Hallucinator* is added. $\mathbb{1}_{[k^- \neq q]} \in \{0,1\}$ is an indicator function, which equals to 1 iff $k^- \neq q$. Notably, we compute the loss for all the positives, including (q,k), (k,q), $(k,\hat{q})$ and $(\hat{q},k)$.

We continue to demonstrate the structure of SimSiam with *Hallucinator* in Figure A.3. Notably, asymmetric extrapolation is applied in embedding space. Therefore, the projector is added before feeding the feature vector k to *Hallucinator*. We define one side of the loss as follows:

$$D(p,k) = -\left(\frac{q}{\|q\|_2} \cdot \frac{k}{\|k\|_2} + \frac{\hat{q}}{\|\hat{q}\|_2} \cdot \frac{k}{\|k\|_2} \right), \tag{A.3}$$

where (p,k) are encoded from two views with and without a projector h respectively. If the encoder is noted as f, then $p = h(f(view1))$ and $k = f(view2)$. Notably, the total loss is symmetric. If we switch two views by defining $k' = f(view1)$ and $p' = h(f(view2))$, we get the final form of loss as follows:

$$\mathcal{L}_{SimSiam} = \frac{1}{2} D(p,k) + \frac{1}{2} (p', k'). \tag{A.4}$$

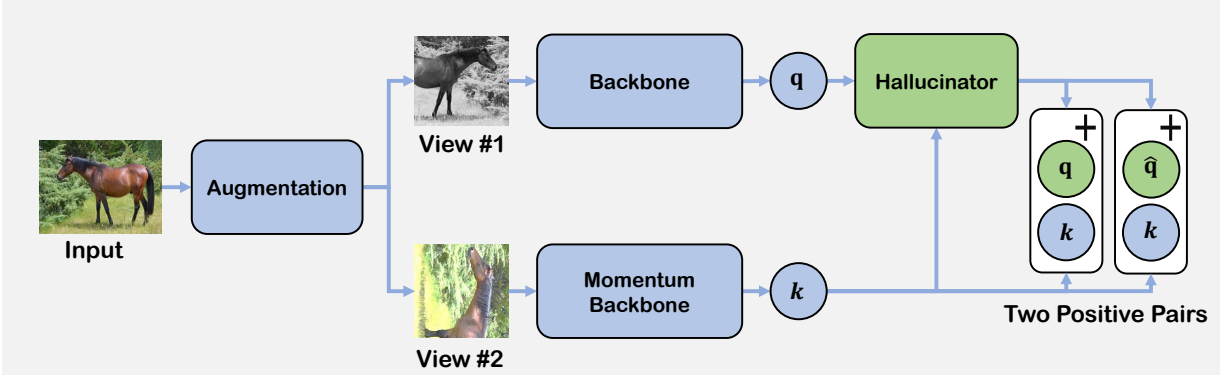

Figure A.1: Illustration of the pipeline based on MoCoV1 [2] with *Hallucinator*.

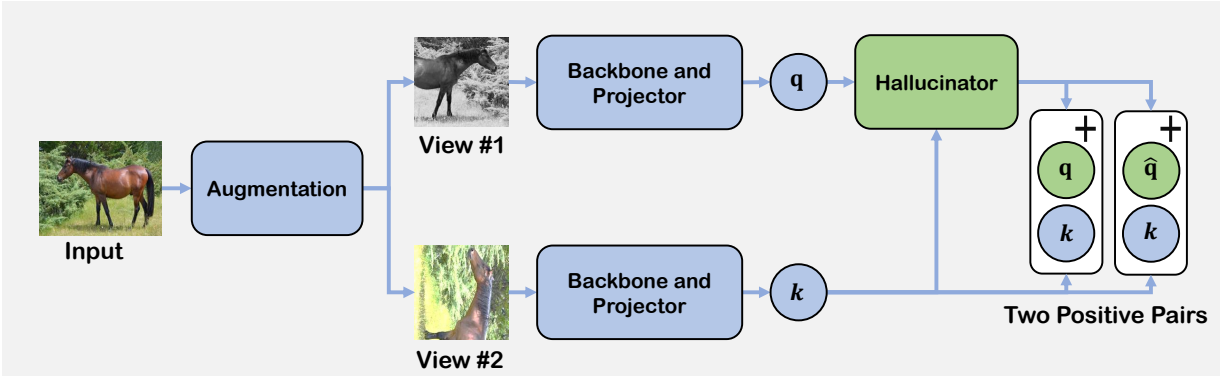

Figure A.2: Illustration of the pipeline based on SimCLR [3] with *Hallucinator*.

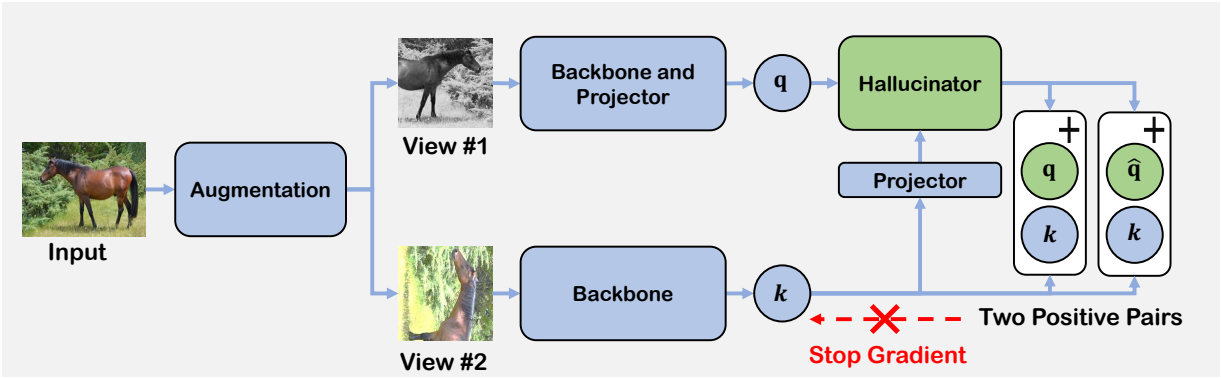

Figure A.3: Illustration of the pipeline based on SimSiam [7] with *Hallucinator*.

A.2 The Center Sampling Method

We visualize the different cropping methods in Figure A.4. Importantly, random cropping sometimes introduces false positive pairs [12], posing a negative influence on overall training. Such adverse influence

is more noticeable when hallucinated sample is generated without sharing mutual semantic meaning with the other positive feature vector. While center cropping reduces the sampling areas and successfully avoids positive pairs, it contains less variance, thus gaining sub-optimal embeddings. If center-suppressed sampling is introduced, the mutual information between positive pairs will be reduced. Then, the benefits of hallucinated samples can be further enhanced.

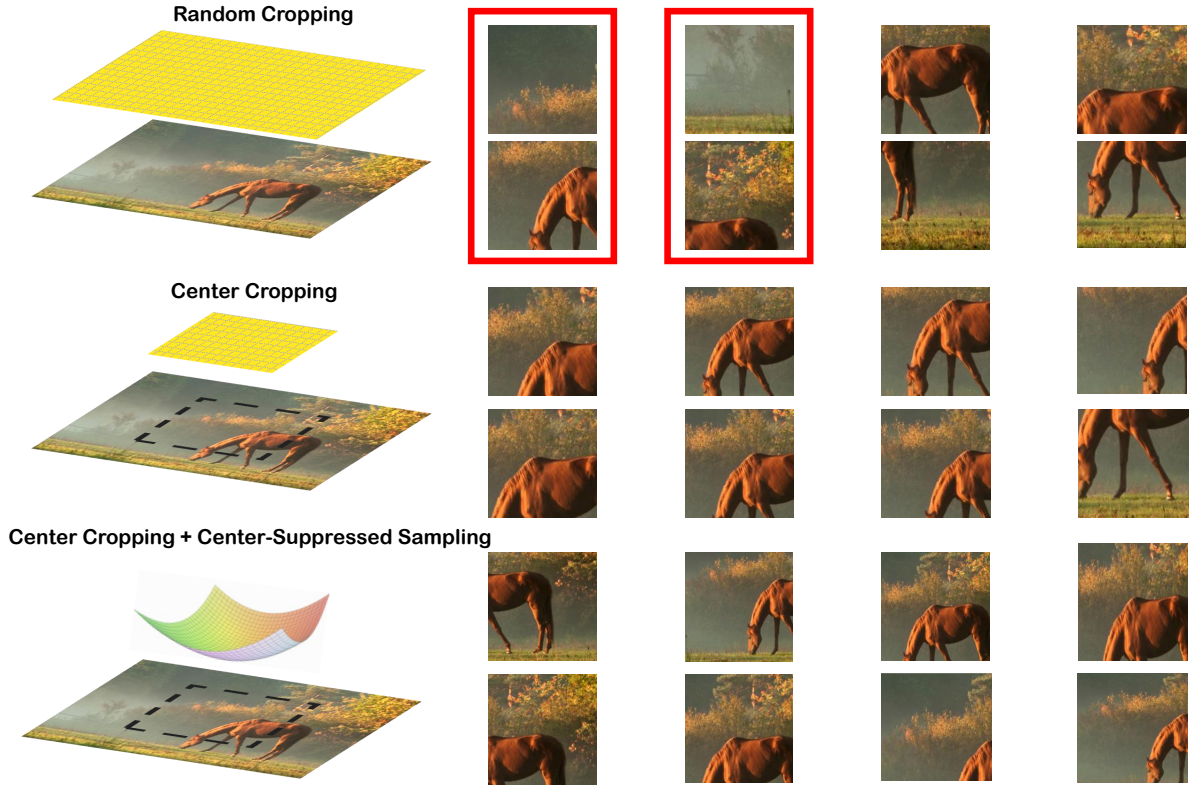

Figure A.4: Visualization of three different cropping methods. We demonstrate the sampling probability (distribution) and operable regions for three settings on the left, and their sampled pairs on the right. Pairs cropped by Random Cropping sometimes introduce false positives, which are highlighted in red boxes. Center Cropping avoids false positives but fails to incorporate sufficient variance. Center Cropping with center-suppressed sampling provides the idea views for *Hallucinator* with enough variance and mutual semantic information.

Appendix B

Extended Agriculture-Vision Dataset

B.1 Agriculture-Vision Dataset

B.1.1 Review of the Original Released Dataset

In the original work of Chiu et al.(2020) [27] choices surrounding the tiling, pre-processing, and data format of the original dataset were largely made to allow for *easy* consumption by users. Specifically,

1. Although there are many images, each image is relatively small, roughly 100KB in size.

2. Images are compressed to JPEG further reducing the data size, and making it easier to move and store.

3. The pre-processing used in the experiments was applied to the released data so the data was "standardized" and results could be more easily reproduced.

4. Binary images corresponding to label layers provides the target in a format requiring minimal manipulation before training and enables the possibility of multiple classes to be present for a single

Table B.1: Flights Per Field

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Flights	1391	394	183	82	39	18	5	3	0	0	0	0	0	1	0

Table B.2: Seasons Per Field

	1	2	3
Seasons	1827	246	43

pixel.

While all of these considerations certainly contributed to an easing of use for reproducing the results of that work, they also tend to limit further studies and modeling options. Instead in the present work, we elect to recreate the full-field images, save labels as polygons as they were originally annotated, and leave the data in its raw form without baked-in pre-processing.

B.1.2 Flights and Fields

While the original work described the data as collected from 3,432 farmlands across the Midwestern United States, the imagery is actually from 3,432 unique *full-field images*, i.e., *"flights"*, over 2,116 farmlands during the 2017-2019 growing period. That is, while most farmlands, i.e., *"fields"*, were annotated only once, about half of the fields were annotated multiple times, potentially within the same season or different seasons. This does not likely impact the modeling approach or structure in any significant way, however, it is important to clarify this language.

The number of flights per field varies from 1 to 14 as seen in Table B.1. Most field imagery is from only a single season, and about 14% are from multiple seasons as seen in Table B.2.

B.1.3 Class labels

Label structure

In the original dataset, the target labels were represented as binary channels, one per class, for each tile. This enabled the targets to be quickly consumed by most modeling frameworks with minimal modification needed. Additionally, it enabled multiple classes per pixel.

As noted in the original work, most pixels have zero annotations, i.e., they are considered 'background'. In fact, 70.14% of the images have no annotations, making the label space largely empty. Of the remaining images, 29.74% have a single class label, 0.13% have two labels, and 0.001% have more than three labels.

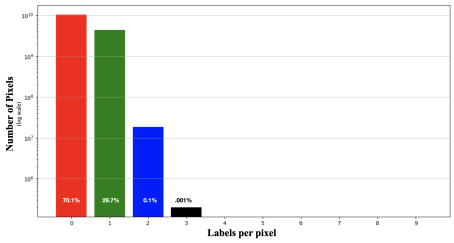

Figure B.1: The number of classes per pixel across the original Agriculture-Vision dataset. The data is plotted on a log scale and the fraction of the total pixels is shown printed in the corresponding bar.

The label distribution can be seen in Figure B.1. We draw attention to this to emphasize how rare multiple labels are. The choice to store each label as a separate channel may appear inefficient given this rarity of overlap.

Therefore, we are releasing the labels as JSON files, where each label is described as a MultiPolygon of pixel coordinates. The original annotations were stored as MultiPolygons of geo-coordinates, but these have been mapped to pixel-coordinates so as to preserve anonymity and enable plotting on the full-field images which also have had their geo-information removed. Each JSON file is on the order of bytes to a few kilobytes, whereas each label channel in the original dataset was on the order of 1kB. The JSON format also provides flexibility in that it places no constraints on overlapping or non-overlapping patterns.

Although these MultiPolygons may appear like a collection of instances, we advise against using them for instance-segmentation tasks as not all labels correspond to meaningful "instances", with labels like nutrient deficiency or drydown corresponding to regions experiencing a certain characteristic.

Label semantics

Images of the nine patterns are provided in [27]; however, descriptions and semantics of the patterns are not included. Additional detail and clarification on the semantics of these annotations are important for understanding the possibility of pattern co-occurrence.

Conceptually, these nine labels are not all mutually exclusive because they correspond to patterns at different semantic levels across the field. For example, crops in a region of the field could be experiencing a nutrient deficiency, a *health* pattern; in that same region, a *mechanical* pattern like double plants may have also occurred. As another example, a weed cluster (which describes *vegetation*) could be located in a pool of water (which describes the *soil*), which is actually quite common. However, because patterns like double plants and planter skips correspond to the spacing of crops, they cannot occur together.

A description of the classes and their relationship to one another are as follows:

Waterways These are *unmanaged* areas of the field that do not include crops (for harvest), but instead contain grasses and other vegetation, often to prevent erosion. Although "weeds" may grow in the waterways (or spread into the field from waterways), they are not labeled as weeds because they are growing in this non-crop region. Waterways are often excluded from the field bounds and therefore only the portion of the waterways included inside the field bounds have been annotated.

Clouds Clouds are the only pattern occurring *above* the ground and not describing the crop or field itself. While certain patterns may be present on the field below the clouds, if they are not visible, only the cloud pattern is annotated.

Water Water is a pattern describing the *ground* or *soil*. Often weeds will grow within pools of water.

Nutrient Deficiency Nutrient Deficiency describes the *health* of the crop and makes no statement about how the crop is arranged in the field. Therefore patterns like a double plant or endrow may also encompass nutrient-deficient crops.

Storm Damage Like nutrient deficiency, storm damage describes the *health* of the crop.

Drydown Drydown describes the *state* of the crop, in this case, the crop in its final phase of development prior to harvest. Like nutrient deficiency, it can overlap with *mechanical* patterns.

Endrow Endrow is a *mechanical* pattern corresponding to the tracks of the planting and spraying equipment. Endrows may overlap with any health or state patterns.

Planter skip This *mechanical* pattern is an unwanted gap in the regular planting of the crop. Weeds could potentially grow in these gaps, although that is uncommon.

Double plant This *mechanical* pattern results from an overlap or over-planting of the crop. Crops in a double plant may also be included in state or health labels as well as endrows.

Weeds Weeds are any plant growing in an undesired location. They can co-occur with mechanical patterns or water. Weeds within waterways are ignored.

B.1.4 Full-Field Imagery

Each raw flight image is quite large- often upwards of 10,000 × 10,000 pixels in dimension and over 1GB in size. Therefore the authors of [27] generated the dataset as 94,986 tiles 512 × 512 in size. While this has the advantage of potentially being more manageable, it does place a limit on the types of models constructed and approaches used.

A full-field image is seen in Figure B.2. This view shows the complexity of the field and how different patterns are organized across it.

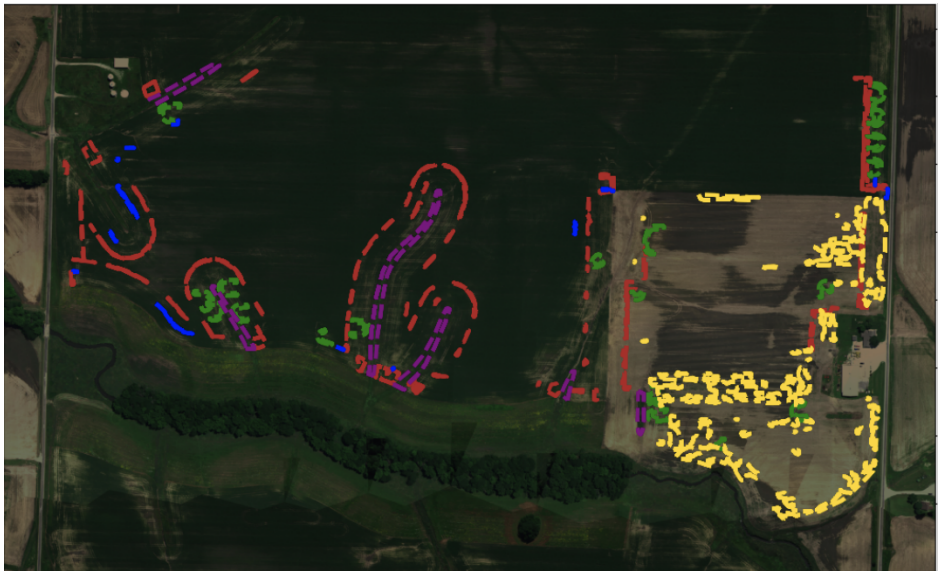

Figure B.2: An image of a full field and associated annotations. Due to the size and resolution of the image, annotations appear as lines but, in fact, are tiny polygons when zoomed in. [Red] double plant, [Green] endrow, [Blue] planter skip, [Purple] waterway, [Yellow] weed_cluster

.

We have seen that using a full-field reconstruction is very useful for training. In the Second Agriculture-Vision Challenge at CVPR 2021, the top team first reconstructed much of the full-field images by stitching the tiles together based upon the pixels at which they were cropped [214]. This enabled them to take many more random crops during training and introduce additional variety into their augmentation protocol. Additionally, they used a full-field reconstruction for inference and averaged the different views of each

pixel. Both tactics produced significant improvement in their results, suggesting that using the full-field image will provide advantages for many different modeling frameworks.

Preprocessing

A key challenge with remote sensing data is the variety of sensors used and the potentially dramatically different response curves they may have. Data from 2017 was captured with a sensor returning values between 0 and 255. In contrast, data in 2018 and later was collected as int16 channels with values ranging between 0 and 65,536. To address this, the original data was clipped between the following bounds:

$$\begin{aligned} V_{lower} &= \max(0, p_5 - 0.4 \times (p_{95} - p_5)) \\ V_{upper} &= \min(255, p_{95} + 0.4 \times (p_{95} - p_5)) \end{aligned} \tag{B.1}$$

where p_5 and p_{95} are the 5^{th} and 95^{th} percentile pixel values for each of the four channels in a given tile; pixels corresponding to invalid regions were excluded from this calculation. While creating a level of normalization, this has the effect of making the images look artificial (pink/cyan/lime/purple) in some cases. Additionally, this makes incorporating new sources of data challenging as the new data must be mapped back in the same manner.

Normalization can provide performance and (training) speed gains. However, by releasing the dataset raw, we leave the choice of normalization up to the researcher. Earth observation and remote sensing analysis may bring in data sources from many different sensors over many years, all of which may cause shifts to the underlying distribution. Similarly, because of the seasonality of the agricultural data, global statistics over many images can be shifted based on the timing and frequency of image capture.

Additionally, agricultural data is unique in the meaningfulness of the specific colors in the images [215].

Traditional computer vision algorithms for agriculture often relied on vegetative indices which are a ratio of different channels [216]. In fact, the specific sensor designs used to collect this data were chosen as narrow-band sensors such that these indices could be calculated with a high signal-to-noise ratio. Therefore, normalizing channels independently can break this latent relationship which the sensor design was specifically designed to capture. Furthermore, because much of this data is narrow-band, standard normalization and color-augmentation approaches are often not appropriate.

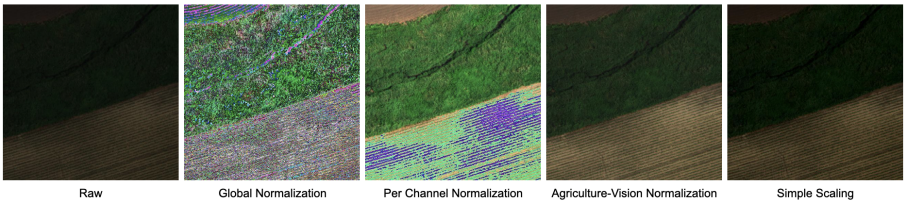

Figure B.3: Pre-processing has a very significant impact on the appearance and statistics of the images. (Raw) Because of the narrow-band nature of the data which predominantly occupies the lower half of the allowed data range, images in their raw form appear dark. (Global Norm) Performing standard normalization on the entire image produces an unusual appearance. (Per Channel Normalization) performing standard normalization on each channel independently also produces an unnatural image and breaks the ratios between the channels which have significance in agriculture. (CVPR Normalization) Normalization of Equation (B.1). (Simple Scaling) Normalization according to the procedure in this section.

Images in their raw form appear dark because they are narrow-band and only tend to occupy the lower half of the int16 range. For visualization purposes, the following procedure (written in python format) for Simple Scaling may be useful.

```
def image_norm(x, b=1):
    y = x[x >= 0]
    x_max = np.percentile(y, 100-b)
    x_min = np.percentile(y, b)
    x = (x-x_min) / (x_max - x_min + 1e-20)
    x = x.clip(0, 1)
```

```
return x

}
```

All of the data we release here, both the supervised dataset and the new unsupervised dataset, is in its raw form as it was collected for that year. 2017 data ranges between 0 and 255 and 2018-2020 ranges between 0 and 65,536; any pixel falling outside of those limits is considered invalid and can be truncated to the appropriate limit or mapped to a separate invalid indicator (e.g. -1) if desired.

The impact of different normalization approaches is of key interest, and by leaving the data in its raw form, we hope to encourage significant research in this area in the future.

B.2 Fine-Grained Segmentation Task

B.2.1 Dataset

A sample of the imagery (combined RGB, as well as individual R-G-B-N channels) and annotations for the downstream fine-grained segmentation task are shown in Figure B.4.

B.3 Additional Results

B.3.1 Balance Factor: Instance-level and pixel-level loss

To optimize the performance of the MoCo-PixPro model, we study different values of the balancing factor of the loss function. We pre-trained the model on 1200 flights and reported the results from the downstream non-linear classification task. Based on our experiments, α is set to 0.4 for all the pre-training.

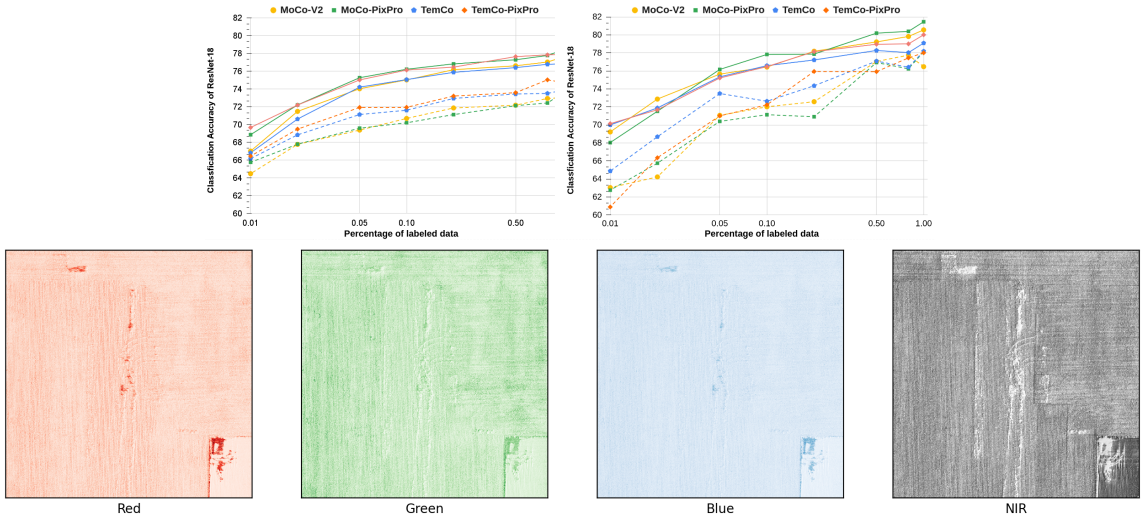

Figure B.4: An example field and associated labels in the field segmentation task. The RGB image (top-left), segmentation labels (top-right), and individual RGBN channels in 16-bit integer format (bottom).

Table B.3: Ablation study on balance factor

Balance Factor α	0.10	0.30	0.40	0.50	0.70
Accuracy	73.26	75.75	**78.38**	77.02	76.04

B.3.2 Downstream Classification: Linear Probing

We show the exact accuracy under the linear classification protocol on labeled Extended Agriculture-Vision in Table B.4, covering different pre-training approaches and backbones, varying from ResNet-18, Resnet50, and Swin-T.

B.3.3 Downstream Classification: Non-Linear Probing

In Table B.5, we continue to show the accuracy under the non-linear classification protocol on labeled Extended Agriculture-Vision. The study on label efficiency is more extensive under this setting. All the weights are pre-trained from 3600 flights of images.

Table B.4: Linear Probing

Weights	Backbone	100.00% labeled data	10.00% labeled data
Random	ResNet-18	62.31	55.34
ImageNet	ResNet-18	68.52	60.87
MoCo-V2	ResNet-18	73.58	62.14
MoCo-PixPro	ResNet-18	**73.74**	**63.75**
TemCo	ResNet-18	73.34	63.55
TemCo-PixPro	ResNet-18	72.24	63.59
Random	ResNet-50	62.13	54.71
ImageNet	ResNet-50	71.22	67.54
MoCo-V2	ResNet-50	77.65	69.59
MoCo-PixPro	ResNet-50	**77.98**	**70.49**
Random	Swin-T	58.39	56.30
ImageNet	Swin-T	64.14	61.61
MoCo-V2	Swin-T	73.29	71.15
MoCo-PixPro	Swin-T	**75.53**	**73.07**
TemCo	Swin-T	74.11	71.92

B.3.4 Ablation Study: Number of flights for pre-training

We use a ResNet-18 backbone and basic MoCo-V2 for experiments. When the number of flights used for SSL is increased from 300 to 3600, we observe stable improvement in the downstream classification task under the non-linear probing setting; this gain is confirmed regardless of the fraction of labeled dataset for tuning (Table B.6).

Ablation Study: Reduced Flights, Non-Linear Probing

We confirm that this result holds across SSL modules by examining performance using 1200 vs. 3600 flights for pre-training and evaluating under the non-linear probing paradigm for AV+ classification (Table B.7).

End-to-End Fine-Tuning

We report the end-to-end accuracy under the end-to-end classification protocol on labeled Extended Agriculture-Vision (AV+) in Table B.8. Results cover different pre-training approaches and backbones.

Table B.5: Non-Linear Probing on 3600 Flights

Weights	Backbone	100.00% labeled data	80.00% labeled data	50.00% labeled data	20.00% labeled data
Random	ResNet-18	62.92	62.81	62.01	61.36
ImageNet	ResNet-18	72.25	71.58	71.18	70.54
MoCo-V2	ResNet-18	77.58	77.03	76.60	76.16
MoCo-PixPro	ResNet-18	**78.38**	**77.77**	**77.28**	**76.82**
TemCo	ResNet-18	76.83	76.78	76.40	75.87
TemCo-PixPro	ResNet-18	77.89	77.83	77.04	76.46
Random	ResNet-50	63.49	62.81	62.95	62.26
ImageNet	ResNet-50	75.35	75.15	73.75	72.84
MoCo-V2	ResNet-50	80.62	80.01	79.38	77.93
MoCo-PixPro	ResNet-50	**80.66**	**80.10**	**79.88**	**78.23**
Random	Swin-T	59.88	59.80	59.28	59.09
ImageNet	Swin-T	75.35	75.15	73.75	72.84
MoCo-V2	Swin-T	79.31	78.81	78.56	77.59
MoCo-PixPro	Swin-T	**80.02**	**79.52**	**78.89**	**77.88**
TemCo	Swin-T	79.01	78.24	77.99	77.11

Weights	Backbone	10.00% labeled data	5.00% labeled data	2.00% labeled data	1.00% labeled data
Random	ResNet-18	60.18	59.05	59.00	58.15
ImageNet	ResNet-18	68.72	67.76	65.20	61.79
MoCo-V2	ResNet-18	75.02	74.03	71.48	67.01
MoCo-PixPro	ResNet-18	**76.22**	**75.27**	**72.20**	**69.85**
TemCo	ResNet-18	75.05	74.21	70.62	65.32
TemCo-PixPro	ResNet-18	76.13	75.02	72.19	69.66
Random	ResNet-50	61.42	60.61	59.05	58.10
ImageNet	ResNet-50	71.92	70.86	67.60	63.91
MoCo-V2	ResNet-50	77.66	75.72	74.32	71.58
MoCo-PixPro	ResNet-50	**77.92**	**76.77**	**75.52**	**71.61**
Random	Swin-T	58.33	57.81	57.62	57.08
ImageNet	Swin-T	63.11	62.30	60.44	58.00
MoCo-V2	Swin-T	74.58	68.85	67.70	66.59
MoCo-PixPro	Swin-T	**76.24**	**70.50**	**68.88**	**66.59**
TemCo	Swin-T	74.51	68.62	66.99	65.58

Table B.6: Ablation study on the pre-training size of data for pre-training with MoCo-V2.

Number of Flights	100.00% labeled data	20.00% labeled data	1.00% labeled data
300	71.91	69.30	62.53
1200	74.73	71.86	64.47
3600	77.58	76.16	67.01

Similar results are shown in the setting of weights pre-trained with only 1200 flights in Table B.9.

Table B.7: Non-Linear Probing on 1200 Flights

Weights	Backbone	100.00% labeled data	80.00% labeled data	50.00% labeled data	20.00% labeled data
Random	ResNet-18	62.92	62.81	62.01	61.36
ImageNet	ResNet-18	72.25	71.58	71.18	70.54
MoCo-V2	ResNet-18	73.02	72.93	72.17	71.86
MoCo-PixPro	ResNet-18	73.53	72.42	72.12	71.12
TemCo	ResNet-18	73.85	73.50	73.43	72.93
TemCo-PixPro	ResNet-18	**74.73**	**75.03**	**73.58**	**73.20**
Weights	Backbone	10.00% labeled data	5.00% labeled data	2.00% labeled data	1.00% labeled data
Random	ResNet-18	60.18	59.05	59.00	58.15
ImageNet	ResNet-18	68.72	67.76	65.20	61.79
MoCo-V2	ResNet-18	70.68	69.37	67.77	64.47
MoCo-PixPro	ResNet-18	70.19	69.59	67.79	65.75
TemCo	ResNet-18	71.60	71.12	68.83	66.17
TemCo-PixPro	ResNet-18	**71.94**	**71.92**	**69.48**	**66.45**

B.3.5 Fine-Grained Semantic Segmentation

In this downstream task, we train each model for 30 epochs using the Adam optimizer. We use the one-cycle learning rate with a maximum learning rate of 0.001, minimum learning rate of 1e-7, initial learning rate of 4e-5, cosine annealing, base momentum of 0.85, and maximum momentum of 0.95 [154]. We show performance for each of the four classes of interest under the different training paradigms in the fine-grained semantic segmentation task.

Table B.8: End-to-End Fine-Tuning on 3600 Flights

Weights	Backbone	100.00% labeled data	80.00% labeled data	50.00% labeled data	20.00% labeled data
Random	ResNet-18	65.32	64.91	65.01	64.36
ImageNet	ResNet-18	80.19	80.07	79.03	77.42
MoCo-V2	ResNet-18	80.57	79.84	79.23	78.10
MoCo-PixPro	ResNet-18	**81.47**	**80.40**	**80.21**	78.11
TemCo	ResNet-18	79.10	78.05	78.28	77.23
TemCo-PixPro	ResNet-18	80.02	79.01	78.97	**78.15**
Random	ResNet-50	78.42	77.17	76.15	71.55
ImageNet	ResNet-50	81.01	80.94	80.77	80.09
MoCo-V2	ResNet-50	81.68	81.55	81.39	81.05
MoCo-PixPro	ResNet-50	**82.22**	**81.86**	**81.42**	**81.11**
Random	Swin-T	77.70	76.00	75.35	73.56
ImageNet	Swin-T	81.06	80.94	80.77	80.09
MoCo-V2	Swin-T	84.49	84.39	84.12	81.76
MoCo-PixPro	Swin-T	**85.58**	**85.37**	**84.85**	**83.41**
TemCo	Swin-T	84.91	84.55	84.01	81.43

Weights	Backbone	10.00% labeled data	5.00% labeled data	2.00% labeled data	1.00% labeled data
Random	ResNet-18	64.18	64.15	62.58	59.95
ImageNet	ResNet-18	75.45	73.90	64.98	62.90
MoCo-V2	ResNet-18	76.44	75.66	**72.89**	69.24
MoCo-PixPro	ResNet-18	**77.85**	**76.18**	71.54	68.06
TemCo	ResNet-18	76.60	75.35	71.90	70.01
TemCo-PixPro	ResNet-18	76.49	75.23	71.70	**70.16**
Random	ResNet-50	67.48	66.40	64.62	61.60
ImageNet	ResNet-50	78.07	75.36	70.42	66.67
MoCo-V2	ResNet-50	79.13	77.08	74.55	71.80
MoCo-PixPro	ResNet-50	**80.14**	**78.06**	**74.95**	**71.86**
Random	Swin-T	70.19	68.84	61.56	55.26
ImageNet	Swin-T	78.07	75.36	70.42	66.67
MoCo-V2	Swin-T	81.04	77.49	75.23	72.09
MoCo-PixPro	Swin-T	**81.15**	**78.46**	**76.53**	**73.88**
TemCo	Swin-T	80.85	77.51	75.33	72.40

Table B.9: End-to-End Fine-Tuning on 1200 Flights

Weights	Backbone	100.00% labeled data	80.00% labeled data	50.00% labeled data	20.00% labeled data
Random	ResNet-18	65.32	64.91	65.01	64.36
ImageNet	ResNet-18	**80.19**	**80.07**	**79.03**	**77.42**
MoCo-V2	ResNet-18	76.48	77.77	77.05	72.59
MoCo-PixPro	ResNet-18	78.13	76.25	76.94	70.93
TemCo	ResNet-18	78.20	76.42	77.10	74.36
TemCo-PixPro	ResNet-18	78.02	77.46	75.93	75.95
Weights	Backbone	10.00% labeled data	5.00% labeled data	2.00% labeled data	1.00% labeled data
Random	ResNet-18	64.18	64.75	62.58	59.95
ImageNet	ResNet-18	**75.45**	**73.90**	64.98	62.90
MoCo-V2	ResNet-18	72.03	71.08	64.24	63.05
MoCo-PixPro	ResNet-18	71.15	70.41	65.77	62.77
TemCo	ResNet-18	72.65	73.50	**68.70**	**64.89**
TemCo-PixPro	ResNet-18	72.25	71.02	66.37	60.91

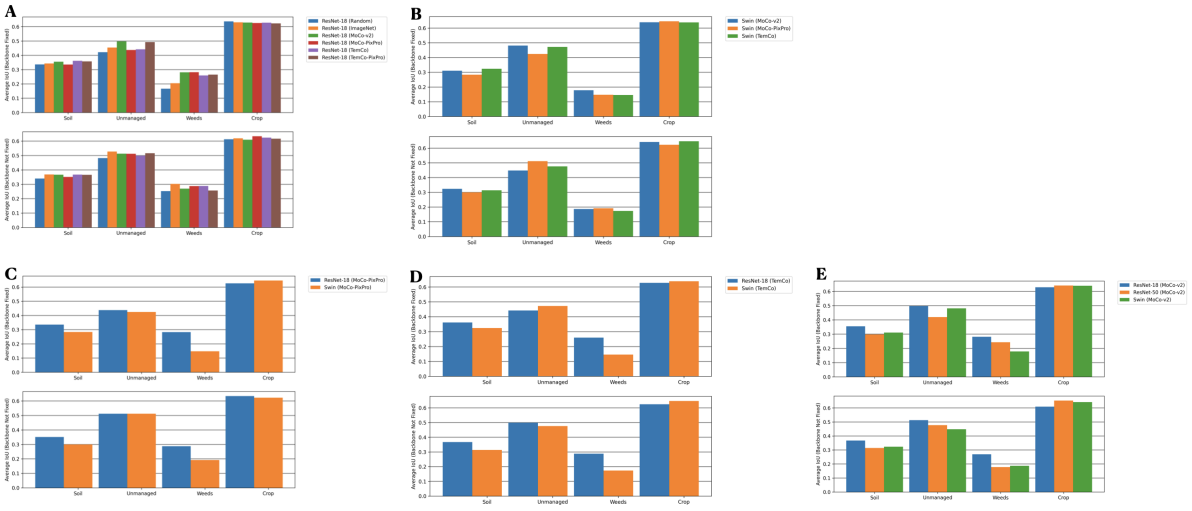

Figure B.5: (A) Average IoU in each class over the test dataset for each model in the field segmentation task with a fixed backbone (top) and fine-tuned backbone (bottom). Each model architecture uses the ResNet-18 encoder. (B) Average IoU in each class over the test dataset for each model in the field segmentation task with a fixed backbone (top) and fine-tuned backbone (bottom). Each model uses the same weight initialization from MoCo-V2 pre-training, but the encoder architecture is different between the models. (C) Average IoU in each class over the test dataset for each model in the field segmentation task with a fixed backbone (top) and fine-tuned backbone (bottom). The two models compared here use weights initialized from MoCo+PixPro pre-training but have different architectures. (D) Average IoU in each class over the test dataset for each model in the field segmentation task with a fixed backbone (top) and fine-tuned backbone (bottom). The two models compared here use weights initialized from TemCo pre-training but have different architectures. (E) Average IoU in each class over the test dataset for each model in the field segmentation task with a fixed backbone (top) and fine-tuned backbone (bottom). Each model architecture uses the Swin Transformer encoder but has different weight initializations.